

## الفصل الأول " البيانات Data "

### مقدمة:

بعد أن تعلمنا فيما سبق كيفية تشغيل برنامج الفيجوال بيزيك وإعداد واجهة البرنامج عن طريق وضع الأدوات على النموذج و ضبط الخصائص من خلال نافذة الخصائص أو برمجيا عن طريق تخصيص قيم للخصائص من خلال نافذة الكود اتضح لنا أنه يمكن للمستخدم أن يُدخل قيما (بيانات) ذات أنواع مختلفة . وفي الواقع العملي عند تصميم برنامج معين ستحتاج لبيانات محددة، فمثلاً عند تصميم برنامج لمدرسة ستحتاج برنامج لتسجيل أسماء الطلاب وتواريخ ميلادهم وأرقام شهادات الميلاد ونتائجهم في السنوات السابقة وساعات الدراسة وبعض البيانات الأخرى، لذلك فأنواع البيانات التي قد تحتاجها في مثل هذا البرنامج ستختلف ، أما عند تصميم برنامج لسوق البورصة فالأمر سيتغير لأننا سنهتم بالأرقام بشكل كبير جدا لان الأرقام في البورصة تعني الكثير بل سنهتم بالكسور الصغيرة جداً . في الحقيقة كل نوع من أنواع البيانات له وزن أو حجم معين على الذاكرة فإذا استخدمنا النوع الصحيح للبيانات فإن ذلك سيوفر علينا جزء من الذاكرة خاصة في البرامج الكبيرة أو البرامج التي تقوم بعمليات حسابية معقدة . فمثلاً في واجهة البرنامج

التالية عندما يقوم المستخدم بادخال :

- اسمه : فإنه قد أدخل قيمة نصية ( عبارة عن نص )
- تاريخ ميلاده : فقد أدخل قيمة عبارة عن تاريخ
- نوعه : فقد أدخل قيمة منطقية ( إما ذكر وإما أنثى )
- عدد أبنائه : فقد أدخل قيمة رقمية .

**ملحوظة هامة :**

كل هذه القيم ( البيانات ) يتم تخزينها في ذاكرة الكمبيوتر RAM بطريقة معينة تختلف حسب نوع هذه القيمة (البيان) . وذلك يرجع إلى الاستخدام الأمثل لذاكرة الكمبيوتر من حيث المساحة التخزينية لأن كل بيان له مساحة تخزينية معينة تختلف على حسب نوع البيان وكذلك العمليات الحسابية والمنطقية التي ستجرى على هذه البيانات .

**مثال واقعي :** إذا أراد مجموعة طلاب عددهم ٢٢ طالب الذهاب إلى رحلة إلى الأهرامات فإنه من الأفضل لهم أن يستقلوا أتوبيس ٢٥ راكب وليس أتوبيس ٥٠ راكب وإن كان في الحالتين سوف يؤدي نفس الغرض ولكن استخدامهم أتوبيس ٢٥ راكب سوف يكون توفير في التكاليف أفضل من أتوبيس ٥٠ راكب ، هكذا تكون ذاكرة الكمبيوتر .

## أنواع البيانات : Data Type

توفر لغة ( V.B .Net ) العديد من أنواع البيانات ( Data Type ) التي تستخدم في تخزين قيم البيانات التي يدخلها المستخدم أو الناتجة من تنفيذ أوامر معينة في البرنامج . ومن أهم أنواع البيانات :

**أولاً : البيانات الرقمية ( Numeric Data Type ) :** وتستخدم في تخزين الأرقام التي تدخل في عمليات حسابية . وتنقسم إلى :

- **البيانات الرقمية الصحيحة ( Integral Numeric Type ) :** وهي التي يمكن أن تخزن بها أرقام ( أعداد ) صحيحة . ومنها :

نوع البيان	Byte	Short	Integer	Long
الحجم	1 بايت	2 بايت	4 بايت	8 بايت

- **البيانات الرقمية الغير الصحيحة ( Nonintegral Numeric Type ) :** وهي التي يمكن أن تخزن بها أرقام بعلامات عشرية . ومنها :

نوع البيان	Single	Double	Decimal
الحجم	4 بايت	8 بايت	16 بايت

**ثانياً : البيانات الحرفية ( Character Data Type )** وتستخدم في تخزين الحروف والكلمات والرموز والأرقام التي لا تدخل في عمليات حسابية مثل رقم الهاتف ومنها :

نوع البيان	Char	String
الحجم	2 بايت	$2^{10}$ بايت

**ثالثاً : البيانات المتنوعة ( Miscellaneous Data Type )** وهي التي لا تندرج تحت البيانات الرقمية والحرفية وتستخدم في تخزين بيانات منطقية أو تاريخ أو غيرها ومنها :

نوع البيان	Boolean	Date	Object
الحجم	2 بايت	8 بايت	4 بايت

## ملحوظة هامة : كل نوع بيان له :

- **حيز تخزين في ذاكرة الكمبيوتر** ( أى عدد من البايت Bytes وهو حجم الذاكرة التى يحتاجها على حسب نوعه )  
فمثلاً البيان من النوع Integer عند استخدامه يشغل 4 بايت.
- **مدى ( حدود )** Range معين من القيم التي يمكن تخزينها في هذا النوع فمثلاً النوع Byte حدود القيم المسموح بتخزينها به تبدأ من الرقم صفر وتنتهى بـ 255.

**مثال توضيحي** لو اردنا تخزين العدد الصحيح 200 فإنه يمكن تخزينه في كل من Byte ، Short ، Integer ، Long ولكن الأصح هو تخزينه في النوع Byte لأنه بذلك سيشغل مساحة أقل في ذاكرة الكمبيوتر.

**والجدول التالي يوضح أنواع البيانات بطريقة أخرى**

نوع البيان	نوع البيان	الحجم الذى يشغله فى الذاكرة بالبايت	طبيعة البيان المخزن داخله (المدى)
أرقام صحيحة	Byte	١	من ٠ الى ٢٥٥
	Short	٢	من - ٣٢٧٦٨ الى ٣٢٧٦٧
	Integer	٤	من - ٢١٤٧٤٨٣٦٤٨ الى ٢١٤٧٤٨٣٦٤٧
	Long	٨	أكثر من ٢١٤٧٤٨٣٦٤٧
أرقام عشرية	single	٤	من ٣٨١٠ الى ٣٨١٠
	double	٨	أكبر ٣٨١٠
	decimal	١٦	مبلغ من المال
حروف	char	٢	تخزين حرف واحد
	string	العديد من الفلايا	تخزين سلسلة حرفية من ٠ الى ٢ بليون حرف
قيمة منطقية	boolean	٢	True or False او yes or No
كائن	Object	٤ بايت	كل الأنواع يمكن تخزينها هنا
تاريخ	date	٨	تاريخ من ١/١/٠٠٠١ الى ٣١/١٢/٩٩٩٩

**معرفة نوع البيان :** عندما تقوم بضبط الخصائص من خلال نافذة الخصائص أو برمجياً من خلال نافذة الكود

كما تعلمنا سابقاً من خلال الصيغة التالية ( `controlname.property=value` )  
**ضبط الخصائص : معناه تخصيص قيمة لهذه الخاصية ، وهذه القيمة لها نوع بيان محدد (Data type).**

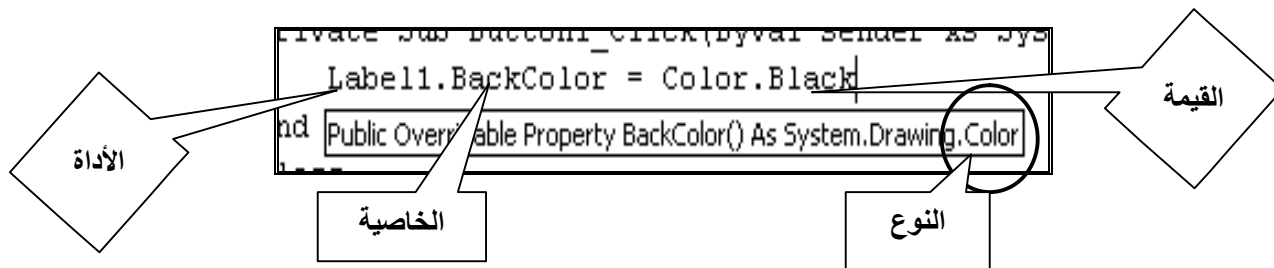
### ولمعرفة نوع البيان :

في نافذة الكود بعد كتابة جملة التخصيص للخاصية قف بمؤشر الفأرة على الخاصية ليظهر مستطيل أصفر به نوع البيان الخاص بهذه الخاصية ويظهر بعد كلمة AS . كما با الشكل التالي :

<pre>Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click     Label1.Text = "تامر محمد عبد المنصف" End Sub Public Overrides Property Text() As String</pre>	الخاصية Text نوعها String
<pre>Label1.Text = "تامر محمد عبد المنصف" Label1.AutoSize = False End Sub Public Overrides Property AutoSize() As Boolean</pre>	الخاصية Auto Size نوعها Boolean
<pre>Label1.AutoSize = False Label1.Width = 120 End Sub Public Property Width() As Integer</pre>	الخاصية Width "عرض الآداة" نوعها Integer

### ملحوظة هامة :

- كما يوجد في لغة الفيجوال بيزك أنواع أخرى ( Types ) فير (Data Types) .
- كما هو الحال بالنسبة للخاصية Back Color للأدوات . كما في الشكل التالي :



- كما أن خصائص الأدوات هي مكان لتخزين البيانات ولها نوع ( Types ) أو نوع بيان (Data Types) .

## المتغيرات Variables

علمت أن أي بيانات يتم التعامل معها تكون مخزنة في ذاكرة الكمبيوتر المؤقتة ( RAM ) والتي تتكون من ملايين من الخلايا المتساوية تسمى بايت .

**والمتغيرات هي**

أماكن محجوزة في ذاكرة الكمبيوتر ( RAM ) ولها نوع بيان ( Data Types ) تخزن بها قيم وهذه القيم يمكن أن تتغير أثناء سير تنفيذ أوامر وتعليمات البرنامج .  
ملحوظة هامة :

- يتم حجز هذه الأماكن في الذاكرة عن طريق الاعلان عنها ( أى اعطاء أسماء للخلايا التي تحمل قيم متغيرة )
- يتم تخصيص قيم للمتغيرات حسب نوع البيان وهذه القيم يمكن أن تتغير أثناء تشغيل البرنامج .
- بمعنى أنه قبل استخدام أى متغير في VB.Net لابد من الاعلان عنه باستخدام الصيغة التالية :

**كيفية الاعلان عن المتغيرات :**

( نوع البيان Data Type ) as اسم المتغير Dim

تستخدم الكلمة Dim ثم كتابة اسم المتغير ثم كلمة AS ثم نوع البيان Data Type كما بالمثال التالي :  
Dim Number1 As Integer

**قواعد تسمية المتغيرات Variables في الفيجوال بيزك:**

- عند اختيار أسماء للمتغيرات لابد من اتباع الآتي حتى يعمل البرنامج بطريقة سليمة :
1. يجب أن يبدأ اسم المتغير بحرف من الحروف الإنجليزية الكبيرة أو الصغيرة أو ( علامة الشرطة السفلية underscore ) .
  2. لا يسمح أن يحتوى الاسم على أى رموز أو علامات خاصة كالمسافة والنقطة وعلامة الاستفهام وعلامة التعجب أو النجمة أو الشرطة أو غيرها
  3. يمكن أن يتكون الاسم من حروف وأرقام وعلامة underscore الشرطة السفلية .
  4. لا يسمح أن يكون الاسم من الكلمات المحجوزة في vb.net مثل: (AS ,Single,me,end ,dim) . وغيرها من الكلمات المحجوزة .
  5. من الأفضل أن يدل الاسم على محتواه .

**أمثلة لأسماء متغيرات صحيحة :**

Y\_name , \_\_tamerMoh , FirstName , Ahmed2012 , Price ,  
U\_birthDate , U\_Gender , \_Name

## أمثلة لأسماء متغيرات غير صحيحة :

السبب	الإسم
لأنه يبدأ برقم ولا بد أن يبدأ بأحد حروف الانجليزية	1stName
يحتوى على مسافة ولا بد ألا يحتوى الاسم على مسافة	Ahmed 2012
يبدأ بالشرطة العادية ولا يسمح بها	-tamerMoh
من الكلمات المحجوزة فى الفيچوال بيزيك ولايسمح بها	As , Single

## اختيار نوع البيان Data Type

عند اختيار نوع البيان Data Type المناسب لكل متغير يجب مراعاة الآتي :

- ماهو حجم البيانات التي سوف تخزن فى المتغير .
- معرفة أو تحديد نوع بيان القيم التي سوف يتم تخزينها في المتغير بناء على العمليات الحسابية والمنطقية التي من الممكن اجراؤها عليه .

## تدريب رقم 1 الإعلان عن المتغيرات Variables

قم باعداد نافذة النموذج التالية التي تمكن المستخدم من أن يدخل بياناته

وهذا الجدول يوضح القيم المتوقع ادخالها من المستخدم ونوعها والتسمية المقترحة :

القيم	نوع البيان Data Type	التسمية المقترحة للمتغير
إسم المستخدم	نصى String	UserName
تاريخ الميلاد	تاريخى Date	UserBirthDate
النوع	منطقى Boolean	UserGender
عدد أفراد الأسرة	رقمى صحيح Integer	UserNoFamily
رقم التليفون	نصى String	UserTelehoneNo

ثم اضغط على زر **تسجيل** مرتين متتاليتين لفتح نافذة الكود الخاصة به و إضافة معالج الحدث Event Handler

وهو ما يطلق عليه أيضاً إجراء الحدث Event Procedure وقم بالإعلان عن المتغيرات كما بالشكل التالي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
        Dim UserName As String
        Dim UserBirthDate As Date
        Dim UserGender As Boolean
        Dim UserNoFamily As Integer
        Dim UserTelefoneNo As String
    End Sub
End Class
```

الإعلان عن المتغيرات

ملحوظة هامة :

- يمكنك الإعلان عن أكثر من متغير باستخدام أمر Dim واحد كما يلي :

```
Dim U_Name As String , U_B_D As Date , U_Gender As Boolean
```

```
Dim X , Y , Z As Integer
```

- بالإعلان عن المتغيرات يتم حجز أماكن في ذاكرة الكمبيوتر RAM حتي يمكن لنا تخزين مدخلات مستخدم البرنامج فيها . ويتم التخزين باستخدام جملة التخصيص .

### جملة التخصيص Assignment :

إن أهم دور للمتغيرات في البرنامج هو تخزين البيانات بها ولكي يتم تخزين قيمة في متغير نستخدم جملة التخصيص .  
وهي جملة من طرفين بينهما علامة (=) وهذه العلامة لا يقصد بها علامة تساوي المعروفة في الحساب ولكن يطلق عليها معامل التخصيص

والطرف الأيسر لهذه العلامة هو المكان الذي توضع فيه القيمة أما الطرف الأيمن عبارة عن القيمة

( VariableName = Value )

والصيغة العامة لهذه الجملة هي

الطرف الأيمن	معامل التخصيص	الطرف الأيسر
قيمة مجردة - قيمة من خاصية - قيمة من متغير - قيمة من تعبير حسابي	=	المتغير أو الخاصية ويعتبر المكان الذي تخزن فيه القيمة

حيث تسمي علامة (=) بمعامل التخصيص . أي أنه يتم حجز مكان في الذاكرة يسمى باسم المتغير وتوضع بداخله القيمة

**أمثلة توضح عملية التخصيص :**

الكود	شرح الكود
Dim Number As Integer Number = 5	الإعلان عن متغير Number نوعه Integer تم تخصيص القيمة المجردة (5) للمتغير Number
Dim Number1 As Integer Dim Number2 As Integer Number1 = 5 Number2 = Number1	الإعلان عن متغير Number1 نوعه Integer الإعلان عن متغير Number2 نوعه Integer تم تخصيص القيمة المجردة (5) للمتغير Number1 تم تخصيص قيمة المتغير Number1 للمتغير Number2
Dim Name As String Name = Label1.Text	الإعلان عن متغير Name نوعه String تم تخصيص قيمة الخاصية Text لأداة Label1 للمتغير Name
Dim Number As Single Number = 7 Number = Number / 3	الإعلان عن متغير Number نوعه Single تم تخصيص القيمة المجردة (7) للمتغير Number تم إعادة عملية التخصيص للمتغير Number من خلال تعبير حسابي
Dim Number As Single = 5.6	تم الإعلان عن متغير Number نوعه Single وتم تخصيص القيمة (5.6) أثناء الإعلان عن المتغير Number في خطوة واحدة.

وفي حالة الإعلان عن متغير حرفي وعدم اعطاؤه قيمة مبدئية في نفس الخطوة يعطيه البرنامج تلقائياً القيمة " " ومعناها سلسلة حرفية فارغة ( نص فارغ ) .

وفي حالة الإعلان عن متغير عددي وعدم اعطاؤه قيمة مبدئية في نفس الخطوة يعطيه البرنامج تلقائياً القيمة صفر .  
**ملحوظة هامة :** بما سبق نستنتج الآتي :

- يمكن تخصيص القيم للمتغيرات أثناء الاعلان عنها في نفس الخطوة أو في مرحلة لاحقة " تالية " .
- يمكن إعادة تخصيص القيم للمتغيرات أي عدد من المرات حسب متطلبات البرنامج ولذلك سميت بالمتغيرات لأن قيمتها تتغير أثناء سير تشغيل البرنامج . مع العلم أن القيمة الجديدة تحل محل القيمة القديمة .
- يمكن أن يكون التخصيص بقيمة مجردة أو بقيمة متغير أو بقيمة تعبير حسابي أو بقيمة خاصية أداة تحكم
- مفهوم العلامة ( = ) في أمر التخصيص يختلف تماماً عن مفهومها في المعادلات الرياضية .
- فعلى سبيل المثال : (  $X = X + 1$  ) فمقبول كأمر تخصيص حيث ( X ) الموجودة على يسار علامة التخصيص يعتبر المكان الذي تخزن فيه القيمة و ( X ) الموجودة على يمينها يمثل القيمة حيث إذا افترضنا أن قيمة ( X ) = 5 فبعد التخصيص يصبح المتغير ( X ) الموجود على يسار العلامة يساوي ( 1 + 5 ) أي ( 6 ) .

**تدريب رقم 2 تخصيص قيم للمتغيرات Variables**

سوف نقوم الآن بتخصيص قيم للمتغيرات التي سبق الاعلان عنها في التدريب رقم 1 والتي تمثل مدخلات المستخدم . وهذه المتغيرات وهي Username - UserBirthDate - UserGender - UserTelephoneNo - UserNoFamily



وفي هذه الحالة سوف نستخدم خصائص الأدوات التي استقبلت مدخلات المستخدم دون تغيير قيمة الخاصية Name للأدوات : وهذه الأدوات هي :

أداة التحكم	الخاصية
TextBox1	Text
TextBox2	Text
TextBox3	Text
RadioButton1	Checked
TextBox4	Text

والآن سوف نقوم بفتح نافذة الكود الخاصة بزر تسجيل واستكمال الكود عن طريق تخصيص قيم للمتغيرات كالتالي :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    Dim UserName As String
    Dim UserBirthDate As Date
    Dim UserGender As Boolean
    Dim UserNoFamily As Integer
    Dim UserTelefoneNo As String
```

الإعلان عن المتغيرات

```
    UserName = TextBox1.Text
    UserBirthDate = TextBox2.Text
    If RadioButton1.Checked = True Then
        UserGender = True
    Else
        UserGender = False
    End If
    UserNoFamily = TextBox3.Text
    UserTelefoneNo = TextBox4.Text
End Sub
```

تخصيص قيم للمتغيرات

" شرح الكود السابق تخصيص قيم للمتغيرات "

- تم تخصيص قيمة للمتغير UserName من الخاصية Text للأداة TextBox1 .
- تم تخصيص قيمة للمتغير UserBirthDate من الخاصية Text للأداة TextBox2 .
- تم اختبار الخاصية ( Checked ) للأداة ( RadioButton1 ) باستخدام جملة IF فإذا كانت قيمة الخاصية Checked تساوي True فإنه يخصص للمتغير UserGender القيمة True أما إذا كانت قيمة الخاصية Checked تساوي False يخصص للمتغير UserGender القيمة False .
- تم تخصيص قيمة للمتغير UserNoFamily من الخاصية Text للأداة TextBox3 .
- تم تخصيص قيمة للمتغير UserTelefoneNo من الخاصية Text للأداة TextBox4 .

ولكن هناك اختلاف بين نوع بيان الخصائص ونوع بيان المتغيرات المستخدمة  
كما هو موضح بالجدول التالي :

أداة التحكم والخاصية	نوع بيان الخاصية	المتغير	نوع بيان المتغير	مدى التوافق أو الاختلاف
TextBox1.Text	String	UserName	String	متوافق
TextBox2.Text	String	UserBirthDate	String	غير متوافق
RadioButton1.Checked	Boolean	UserGender	Boolean	متوافق
TextBox3.Text	String	UserNoFamily	String	غير متوافق
TextBox4.Text	String	UserTelefoneNo	String	متوافق

وهذا لايحوز ولكن لغة الفيجوال بيزيك تتيح ما يسمى بالتحويل الضمني : وهو

التحويل الضمني هو : إمكانية تحويل القيم إلى نوع بيان متوافق مع المتغير أو الخاصية المخصص لها القيمة

### استخدام المتغيرات Variables :

يمكن استعمال متغير على يمين علامة ( = ) ومتغير على يسارها في نفس الأمر

- حيث إذا وجد المتغير على يسار علامة التخصيص ( = ) هو المكان الذي ستخزن به القيمة
- بينما إذا وجد المتغير على يمين علامة التخصيص فهو يعبر عن القيمة الموجودة بداخله والمراد تخزينها  $Price = Price + 10$

### تدريب رقم 3 استخدام المتغيرات Variables

عزيزي الطالب في هذا التدريب سوف نستخدم أداة عنوان Label في عرض قيم مجموعة المتغيرات التي تم التعامل معها في التدريب ١ ، ٢ والأن قم بإضافة أداة عنوان Label5 إلى نافذة النموذج السابق التعامل معها . ثم قم بضبط خصائصها

الخاصية	القيمة المطلوبة
AutoSize	False
Size , Location	يتم ضبطها بالفأرة
BorderStyle	FixedSingle

ثم قم بفتح معالج الحدث الخاص بزر تسجيل واستكمال الكود عن طريق كتابة الكود التالي :

```
Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & UserGender & vbCrLf & _
UserNoFamily & vbCrLf & UserTelefoneNo
```

## لاحظ :

- الكود السابق عبارة عن أمر تخصيص مجموعة متغيرات كقيمة للخاصية Text لأداة التحكم Label 5 .
- حيث توجد المتغيرات على يمين علامة ( = ) إذن فهي القيمة .
- يفصل بين كل متغير والآخر ( & vbCrLf & ) .
- علامة & تستخدم للربط بين النصوص .
- الكلمة المحجوزة ( vbCrLf ) تستخدم في إنشاء سطر جديد .
- تستخدم علامة ( \_ ) حتي يمكن كتابة سطر الكود على أكثر من سطر في حالة إذا كان سطر الكود (Code) طويل بعض الشيء وذلك لتنظيم وتسهيل قراءة الكود
- ( Me ) تعبر عن نافذة النموذج ( Form ) الحالية .

## وسوف يكون شكل معالج الحدث Click لنز الأمر ( تسجيل ) كما يلي :

```
Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As
    Dim UserName As String
    Dim UserBirthDate As Date
    Dim UserGender As Boolean
    Dim UserNoFamily As Integer
    Dim UserTelefoneNo As String
```

الإعلان عن المتغيرات

```
UserName = Me.TextBox1.Text
UserBirthDate = Me.TextBox2.Text
If Me.RadioButton1.Checked = True Then
    UserGender = True
Else
    UserGender = False
End If
UserNoFamily = Me.TextBox3.Text
UserTelefoneNo = Me.TextBox4.Text
```

تخصيص قيم للمتغيرات

استخدام المتغيرات

```
Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & UserGender & _
vbCrLf & UserNoFamily & vbCrLf & UserTelefoneNo
End Sub
```

وبعد الانتهاء من كتابة الكود قم بتشغيل البرنامج بالضغط على مفتاح F5 من لوحة المفاتيح أو رمز ▶ الموجود في شريط الأدوات ثم قم بادخال البيانات في استمارة التسجيل ثم اضغط على زر تسجيل ولاحظ : ظهور محتوى أدوات التحكم على أداة العنوان Label5 .

**كتابة التعليقات والملاحظات :** تتيح لغة الفيجوال بيزيك امكانية كتابة تعليق داخل نافذة الكود لكي تيسر للقارئ فهم الأكواد. كما يستخدمها المبرمج في تدوين بياناته داخل نافذ الكود.

ويبدأ سطر كتابة التعليقات بكلمة REM او علامة ( ' ) حيث أن مايكتب بعدها لايعتبر كود ويتم تجاهله من قبل المترجم ( Compiler ) عند ترجمة الأكواد كما يلي :

```
Private Sub Button1_Click(ByVal sender As System.Object,
    REM كل ما يبدأ بهذه الكلمة لا يعتبر كود بل يعتبر تعليق
    ' الإعلان عن المتغيرات
    Dim UserName As String
    Dim UserBirthDate As Date
    Dim UserGender As Boolean
    Dim UserNoFamily As Integer
    Dim UserTelefoneNo As String
```

الإعلان عن المتغيرات

```
    REM تخصيص قيم لهذه المتغيرات
    UserName = Me.TextBox1.Text
    UserBirthDate = Me.TextBox2.Text
    If Me.RadioButton1.Checked = True Then
        UserGender = True
    Else
        UserGender = False
    End If
    UserNoFamily = Me.TextBox3.Text
    UserTelefoneNo = Me.TextBox4.Text
```

تخصيص قيم للمتغيرات

```
    ' استخدام هذه المتغيرات
    Me.Label5.Text = UserName & vbCrLf & UserBirthDate & vbCrLf & UserGender & _
        vbCrLf & UserNoFamily & vbCrLf & UserTelefoneNo
End Sub
```

استخدام المتغيرات

## الثوابت Constant :

أماكن محجوزة في ذاكرة الكمبيوتر ( RAM ) ولها نوع بيان ( Data Types ) تخزن بها قيم وهذه القيم لا يمكن أن تتغير أثناء سير تنفيذ أوامر وتعليمات البرنامج . ويتم تخصيص قيمه لها أثناء الاعلان عنها فقط

**ملحوظة هامة :**

- يتم حجز هذه الأماكن في الذاكرة عن طريق الاعلان عنها ( أى اعطاء أسماء للخلايا التي تحمل قيم ثابتة )
- يتم تخصيص قيم للثوابت حسب نوع البيان أثناء الاعلان عنها فقط وليس في مراحل لاحقة .
- القيم التي تخص للثوابت لايمكن أن تتغير نهائياً أثناء تشغيل البرنامج .

## الصيغة العامة للاعلان عن الثوابت هي :

القيمة = نوع البيان As اسم الثابت الكلمة المحجوزة Const

تستخدم الكلمة Const ثم كتابة اسم الثابت ثم كلمة AS ثم نوع البيان Data Type ثم علامة التخصيص ( = ) ثم القيمة كما بالأمثلة التالية :

Const C\_Name As String = "جمهورية مصر العربية"

حيث تم الإعلان عن ثابت C\_Name من النوع String وتخصيص قيمة نصية له أثناء الإعلان عنه.

Const Pi As Single = 22/7

حيث تم الإعلان عن ثابت Pi من النوع Single وتخصيص قيمة رقمية له أثناء الإعلان عنه.

Const BirthDate As Date = # 31/ 12 / 1990 #

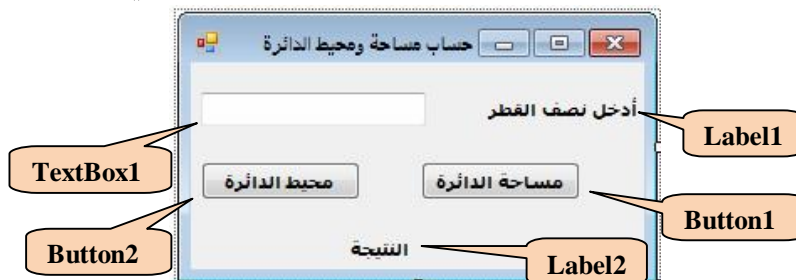
حيث تم الإعلان عن ثابت BirthDate من النوع Date وتخصيص قيمة تاريخ له أثناء الإعلان عنه.

ملحوظة هامة ..

- تستخدم علامتي التنصيص " " مع القيمة النصية حتى تطبع كما هي .
- وتستخدم علامتي # # مع قيم التاريخ والوقت ..

### تدريب رقم 4 الإعلان عن الثوابت :

سنقوم بإعداد نافذة النموذج التالية ( واجهة البرنامج ) كالتالي :



ثم نقوم بفتح نافذ الكود بالضغط على مفتاح F٧ من لوحة المفاتيح وادراج معالج الحدث لكل من زري الأمر Button1 ، Button2 . كما تعلمت سابقاً . كما بالشكل التالي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As
    End Sub

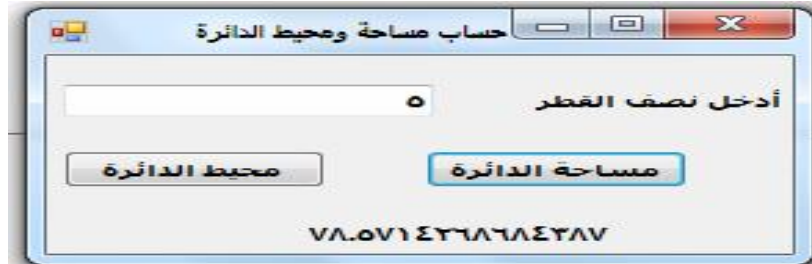
    Private Sub Button2_Click(ByVal sender As
    End Sub
End Class
```

مع العلم أن مساحة الدائرة  $\pi r^2$  ، ومحيط الدائرة  $2\pi r$  . حيث  $r$  هي نصف القطر ،  $\pi$  تساوي ٢٢/٧ .  
ثم قم بكتابة الكود الخاص بحساب مساحة الدائرة في اجراء الحدث للزر Button1 كما يلي :

```
Private Sub Button1_Click(ByVal sender As
    Dim Radius As single
    Const Pi As Single = 22/ 7
    Radius = TextBox1.Text
    Label2.Text = Pi * Radius ^ 2
End Sub
```

## في الكود السابق :

١. تم الاعلان عن متغير Radius من النوع Single .
  ٢. تم الاعلان عن الثابت Pi من النوع Single وتخصيص قيمة ٢٢/٧ له أثناء الاعلان عنه .
  ٣. تم تخصيص قيمة الخاصية Text لأداة التحكم TextBox1 للمتغير Radius .
  ٤. تم تخصيص التعبير الحسابي (  $Pi * Radius^2$  ) للخاصية Text لأداة التحكم Label2 طبقا للقانون الرياضي .
- ثم قم بتشغيل البرنامج بالضغط على مفتاح F5 من لوحة المفاتيح ، ثم أدخل قيمة نصف القطر وانقر على زر مساحة الدائرة :



## تحديد نطاق اعلان المتغيرات Variables والثوابت Constant :

يقصد بنطاق المتغيرات أو الثوابت موضع الإجراء الذي يتم فيه الاعلان عن الثابت أو المتغير واستخدامه .  
قم بكتابة الكود الخاص بحساب محيط الدائرة في نطاق إجراء الحدث للزر Button2. دون الاعلان عن المتغيرات مرة أخرى . كما يلي :

```
Radius = TextBox1.Text
Label2.Text = Pi * Radius * 2
```

ليصبح شكل الكود كما يلي :

```
Private Sub Button1_Click(ByVal sender As Obj
    Dim Radius As Single
    Const pi As Single = 22 / 7
    Radius = TextBox1.Text
    Label2.Text = pi * Radius ^ 2
End Sub

Private Sub Button2_Click(ByVal sender As Obj
    Radius = TextBox1.Text
    Label2.Text = pi * Radius * 2
End Sub
```

- لاحظ: أننا لم نعلن عن المتغير والثابت في نطاق إجراء الزر Button2 ولكن :
- تم الاعتماد على أن المتغير ( Radius ) والثابت ( Pi ) سبق الاعلان عنهما في نطاق إجراء الحدث ( Button1\_Click ) .
- مما يؤدي إلى ظهور مستطيل أصفر عند التوجه بمؤشر الفأرة على الاسم يشير إلى أن المتغير والثابت لم يتم الإعلان عنهما في نطاق الحدث ( Button2\_Click ) رغم أنهما معلنين في نطاق الزر Button1 .
- وبناء على ذلك اتضح لنا أنه لا يمكن استخدام المتغيرات والثوابت إلا في نطاق إعلانها .

## ولحل هذه المشكلة

يمكن الإعلان عن المتغير والثابت مرة أخرى في نطاق إجراء حدث الزر Button2 لأنه لا يمكن استخدامهما إلا في حدود نطاق الإجراء فقط . كما يلي :

```

Start Page Form1.vb Form1.vb [Design]
(General) (Declarations)

Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
        Dim radius As Single
        Const pi As Single = 22 / 7
        radius = TextBox1.Text
        Label2.Text = pi * radius ^ 2
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal
        Dim radius As Single
        Const pi As Single = 22 / 7
        radius = TextBox1.Text
        Label2.Text = 2 * pi * radius
    End Sub
End Class

```

ثم قم بتشغيل البرنامج وأدخل نصف القطر واضغط على زر مساحة الدائرة ومحيط الدائرة لتظهر النتائج .

## الحل الآخر للمشكلة

كما يمكن الإعلان عن المتغيرات والثوابت على مستوى التصنيف Class : وعندها لا نحتاج إلي الإعلان عن المتغيرات عند كل نطاق إجراء حدث .

```

Start Page Form1.vb Form1.vb [Design]
Form1 (Declarations)

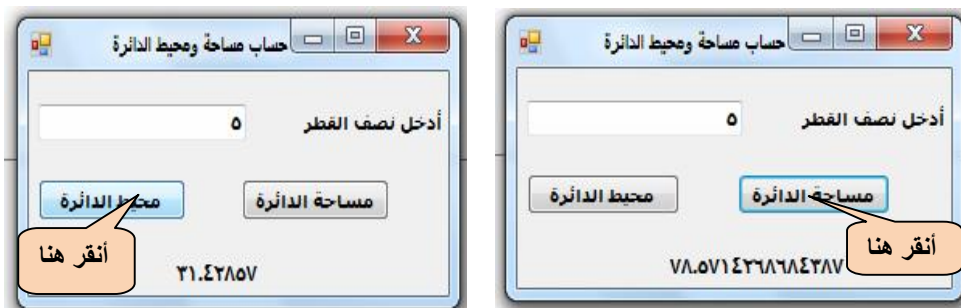
Public Class Form1
    Dim radius As Single
    Const pi As Single = 22 / 7
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal
        radius = TextBox1.Text
        Label2.Text = pi * radius ^ 2
    End Sub

    Private Sub Button2_Click(ByVal sender As System.Object, ByVal
        radius = TextBox1.Text
        Label2.Text = 2 * pi * radius
    End Sub
End Class

```

إعلان المتغيرات والثوابت على مستوى التصنيف Class

ثم قم بتشغيل البرنامج وأدخل نصف القطر واضغط على زر مساحة الدائرة ومحيط الدائرة مرة أخرى لتظهر النتائج .





## الأخطاء في لغة الفيجوال بيزيك :

يجب عند كتابة الكود الالتزام بقواعد اللغة حتى لا نقع في الأخطاء المختلفة : ومنها :

١. الأخطاء الإملائية والنحوية : Syntax Errors :

وهي التي تحدث عند كتابة أي كود بطريقة غير صحيحة مثل كتابة الكلمة Din بدلاً من الكلمة Dim. كما يلي :

Din x As Integer

حيث تم الاعلان عن المتغير x ولكن كتبت الكلمة Dim خطأ.

Const Pi As Single

حيث تم الاعلان عن الثابت Pi ولم تخصص قيمة له أثناء الاعلان عنه.

وللغلب على هذه الأخطاء أنسمخ الفيجوال بيزيك بوجود مثل هذه الأخطاء ونظهر خط أزرق منعرج تحت الخطأ.

٢. الأخطاء المنطقية : Logical Errors :

تحدث عندما نستخدم تعبيرات بطريقة غير سليمة مثل استخدام علامة + لاجراء عملية الضرب .

وهذا النوع من الأخطاء لا يظهر رسائل خطأ ولكن يظهر نتائج غير متوقعة . Label1.Text = X + Radius^2

ولكى يتم التغلب عليها لابد من مراجعة الكود جيداً و اخبار البرنامج باستخدام بيانات معروف قيمة ننتجها مسبقاً .

٣. الأخطاء عند التشغيل : RunTime Errors :

تحدث عند تشغيل البرنامج ونظهر رسالة تحذير عند وجود هذا النوع من الأخطاء ويحدث هذا النوع في حالة :

- تخصيص قيمة أكبر من مدى نوع البيان المستخدم .
- أو تخصيص قيمة نوعها يختلف عن نوع بيان المتغير أو الثابت أو الخاصية ولا يتمكن المترجم من تحويلها إلى هذا النوع .

## ٢ تدريب رقم 5 التعامل مع الأخطاء :

قم باعداد نافذة النموذج التالية و اكتب كود البرمجة في معالج حدث الزر مجموع كما بالشكل التالي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender As System.Object, ByVal e As System.EventArgs) Handles Button1.Click
        ' الإعلان عن متغير لكل مادة
        Dim arabic As Byte
        Dim computer As Byte
        ' تخصيص مدخلات المستخدم لكل متغير
        arabic = Me.TextBox1.Text
        computer = Me.TextBox2.Text
        ' تخصيص ناتج مجموع المتغيرين
        Me.Label3.Text = arabic + computer
    End Sub
End Class
```



لاحظ : في التدريب السابق تم الإعلان عن متغيرين arabic و computer لهما نوع بيان رقمي Byte وهذا النوع يستقبل قيماً تبدأ من 0 وتنتهي عند 255 ومعنى هذا أنه يجب على مستخدم البرنامج أن يلتزم بحدود تلك القيم عند إجراء تخصيص (إدخاله بيانات) لدرجة اللغة العربية والكمبيوتر. وفي حالة عدم إلتزامه بها يحدث خطأ أثناء التشغيل Runtime Errors كما في الحالات التالية :

١. الحالة الأولى : عند تشغيل البرنامج بالضغط على مفتاح F5 من لوحة المفاتيح . بدون ادخال أي بيانات في مربعات النصوص ثم اضغط على زر جمع تظهر لنا رسالة خطأ تشير إلي أنه لا يمكن تحويل القيمة النصية الفارغة إلي نوع البيان Byte وهذا الخطأ يسمى InvalidCastException

اللغة العربية	
الكمبيوتر	

```

تخصيص مدخلات المستخدم لكل متغير
arabic = Me.TextBox1.Text
computer = Me.TextBox2.Text
مجموع المتغيرين
Me.Label3.Text = a

```

InvalidCastException was unhandled  
Conversion from string "" to type 'Byte' is not valid.

وقد حدث ذلك الخطأ . لأننا لم ندخل أي بيانات في درجة اللغة العربية . قم بإيقاف تشغيل البرنامج .

٢. الحالة الثانية : عند تشغيل البرنامج بالضغط على مفتاح F5 من لوحة المفاتيح . ثم ادخال القيمة النصية ( خمسة عشر ) في درجة اللغة العربية ثم اضغط على زر جمع تظهر لنا رسالة خطأ تشير إلي أنه لا يجوز تحويل القيمة النصية ( خمسة عشر ) إلي نوع البيان Byte .

اللغة العربية	خمس عشر درجة
الكمبيوتر	

```

تخصيص مدخلات المستخدم لكل متغير
arabic = Me.TextBox1.Text
computer = Me.TextBox2.Text
مجموع المتغيرين
Me.Label3.Text = a

```

InvalidCastException was unhandled  
Conversion from string "خمس عشر درجة" to type 'Byte' is not valid.

وهذا الخطأ يسمى أيضاً InvalidCastException قم بإيقاف تشغيل البرنامج .

٣. الحالة الثالثة : عند تشغيل البرنامج بالضغط على مفتاح F5 من لوحة المفاتيح ثم كتابة القيمة ٣٠٠ في درجة اللغة العربية ثم اضغط على زر جمع تظهر لنا رسالة خطأ تشير إلي أنه لا يجوز ادخال قيمة اكبر من المسموح به مع نوع البيان الرقمي Byte وهي من ( صفر : ٢٥٥ ) .

اللغة العربية	٣٠٠
الكمبيوتر	

```

تخصيص مدخلات المستخدم لكل متغير
arabic = Me.TextBox1.Text
computer = Me.TextBox2.Text
الفرق بين
Me.Label3.Text = a - b

```

OverflowException was unhandled  
Arithmetic operation resulted in an overflow.

وهذا الخطأ يسمى OverflowException قم بإيقاف تشغيل البرنامج .

والشكل البيضاوي الموضح في الشاشات السابقة يشير إلي نوع الخطأ الذي لم تتم معالجته .

## التغلب على أخطاء التشغيل Runtime Errors :

هناك أكثر من طريقة للتغلب على هذا النوع من الأخطاء :

الطريقة الأولى : استخدام جملة ( Try .....Catch.....End Try )

الطريقة الثانية : التحكم في مدخلات المستخدم بوضع أدوات تحكم معينة وضبط خصائصها بحيث تجبر المستخدم على ادخال بيانات صحيحة

أولاً : الطريق الأولى : استخدام جملة ( Try.....Catch.....End Try )

تستخدم لاكتشاف الأخطاء التي تحدث أثناء التشغيل والتعامل معها :

حيث تتيح للمبرمج التعامل مع الأخطاء بنفسه وإظهار رسائل واضحة لمستخدم البرنامج عند الخطأ .

الفكرة العامة لها : Try

مجموعة الأكواد التي يجب تنفيذها بدون أخطاء والمحتمل حدوث خطأ فيها

Catch ex As Exception

مجموعة الأكواد التي تكتب ولا تنفذ إلا عند حدوث خطأ وغالباً تكون رسالة للمستخدم

End Try

شرح الأمر :

١. يبدأ أولاً تنفيذ الأوامر التي تلي كلمة Try فإذا تم تنفيذها بدون أخطاء يتم تجاهل الأوامر التي تلي

Catch . ثم يتم تنفيذ الأوامر التي تلي End Try .



٢. أما إذا تم تنفيذ الأوامر التي تلي كلمة Try مع وجود أخطاء فلا يستكمل البرنامج بقية الأوامر

التي تلي Try بل ينتقل إلي الأوامر التي تلي Catch وحتى End Try والتي غالباً ما تكون رسالة

للمستخدم .

**ثانياً : الطريق الثانية :** استخدام الأدوات التي تتحكم في مدخلات المستخدم :

مثل ( Combo Box , Radio Button , Check Box ) . حيث تجبر المستخدم على اختيار المدخلات بدلاً من كتابتها داخل أداة TextBox .

- في هذه النافذة لا يمكن أن يختار المستخدم دولة ليست في القائمة .
- كما أنه أيضاً سيختار تاريخ الميلاد من عناصر القائمة بدلاً من كتابته حتى لا يدخل بيانات غير سليمة

### أولويات تنفيذ العمليات الحسابية :

العمليات الحسابية التي يمكن إجراؤها في VB.Net هي :

المعامل ( المؤثر )	العملية
^	الرفع إلى الأس
*	الضرب
/	القسمة العادية
\	قسمة أعداد صحيحة (صحيح القسمة)
Mod	باقي القسمة
E	الجمع
—	الطرح

عند تخصيص تعبير حسابي فإنه قد يحتوي على أكثر من عملية حسابية : فكيف يتم تنفيذها ؟

يتم تنفيذ العمليات الحسابية في لغة VB.Net بالترتيب التالي تنازلياً :

١ - فك الأقواس من الداخل إلى الخارج ( ( ) )

٢ - الرفع إلى الأس ^

٣ - الضرب \* والقسمة العادية / أيهما أسبق من اليسار إلى اليمين

٤ - قسمة الأعداد الصحيحة \ ( صحيح القسمة )

٥ - باقي القسمة Mod

٦ - الجمع E والطرح — أيهما أسبق من اليسار إلى اليمين

• أمثله :

$$5 + 3 * 2 = 11$$

$$(5 + 3) * 2 = 16$$

في المثال الأول سيتم تنفيذ عملية الضرب أولاً ثم الجمع

أما في المثال الثاني سيتم تنفيذ عملية الجمع أولاً لأنها بين قوسين ثم عملية الضرب

- عند إنشاء تعبيرات حسابية لاستخدامها في جملة التخصيص يجب مراعاة هذه الأولويات والالتزام بها حتى لا تحدث أخطاء منطقية . تؤدي إلى نتائج خطأ ....

## الفصل الثاني (التفرع Branching)

درست في خرائط التدفق Flow Chart وخطوات الحل أنه يمكن التفرع واتخاذ القرار باختيار تنفيذ مجموعة خطوات بناء على سؤال معين ، وسوف نجد أن كتابة الكود Code الخاص بالتفرع ما هي إلا خطوات الحل Algorithm ولكن مع الالتزام بقواعد اللغة المستخدمة وطريقة صياغة تعبير التفرع برمجياً :

**أولاً : التفرع باستخدام جملة If .. Then...**

وهي من أوامر التفرع وتستخدم في حالة وجود شرط له بديل واحد فقط أو احتمال واحد فقط هو الاحتمال True (أحادية الاختيار أو Single Selection).

الصيغة العامة لهذه الجملة

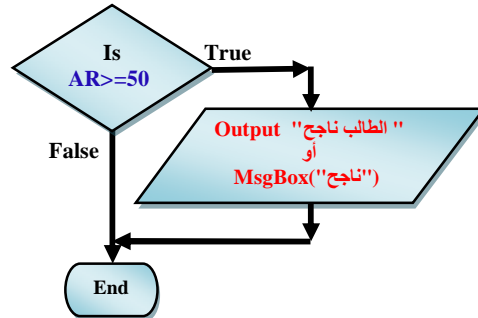
If **Conditional Expression** (التعبير الشرطي أو الشرط) Then  
**Code**  
 End If

الأوامر التي ستنفذ في حالة ما يكون True ناتج هذا الشرط صواب

ما يكتب بعد If هو الشرط أو التعبير الشرطي وفي حالة ما يكون ناتج هذا الشرط صواب True يتم تنفيذ الأوامر التي تلي Then حتى الوصول إلى End If ثم يتم تنفيذ الأوامر التي تلي End If ، وفي حالة ما يكون ناتج هذا الشرط خطأ False لا ينفذ أى شئ من جملة If بل يتم تنفيذ الأوامر التي تلي End If

**مثال على التفرع باستخدام جملة If ..... Then .....**

If **X >= 50** Then  
**MsgBox("ناجح")**  
 End If



**التعبيرات الشرطية Conditional Expression :**

هو جزء من كود Code البرمجة يكون **دائماً** إما صواب True أو خطأ False وذلك بناء على قيمة خاصية Property أو متغير Variable أو بيان آخر بالبرنامج (قيمة مجردة) .

**معاملات المقارنة :**

المعامل	المعنى
>	أكبر من
<	أصغر من
>=	أكبر من أو يساوى
<=	أصغر من أو يساوى
=	يساوى
<>	لا يساوى

**ناتج التعبيرات الشرطية دائماً ما يكون إما صواب True أو خطأ False ويتوقف على العلاقة بين الطرفين.**

• لا يجوز أن نقارن بين قيم أنواع بيانات مختلفة قد لا يستطيع المترجم Compiler تحويلها إلى أنواع بيانات متوافقة مثل  $5 = "Aly"$  ولكن يجوز إجراء هذه المقارنة  $5 = "5"$ ، حيث يستطيع المترجم Compiler تحويل النص "5" إلى نوع بيان عددي متوافق مع القيمة العددية 5 ثم يقوم بالمقارنة وهذا ما يسمى بالتحويل الضمني.

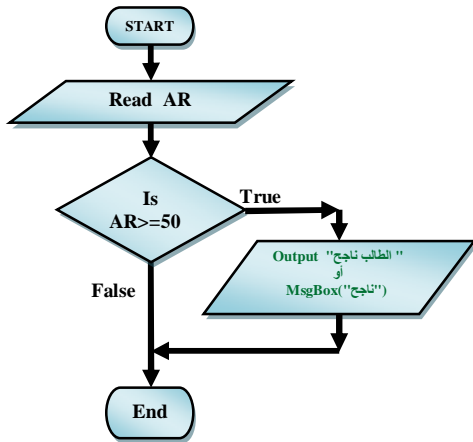
- في الكود If X = 50 Then المعامل يساوي = ليس معامل تخصيص ولكنه معامل مقارنة .
- في الكود X = 50 المعامل يساوي = هو معامل تخصيص

**أمثلة على التعبيرات الشرطية وناتجها :**

التعبير الشرطي	الناتج
$100 < > 100$	False حيث أن القيمتين متساويتين
$300 < > 100$	True حيث أن القيمتين غير متساويتين
Degree = TextBox1.Text	True إما إذا كان قيمة الخاصية Text للأداة TextBox1 تساوي محتوى المتغير Degree أو False إذا كان قيمة الخاصية Text للأداة TextBox1 لا تساوي محتوى المتغير Degree

• تدريب ١ :

قم بتحويل خطوات الحل و خريطة التدفق التالية كما درستها في الفصل الدراسي الأول لعمل برنامج لعملية قراءة درجة طالب في مادة اللغة العربية لمعرفة ما إذا كان ناجحاً علماً بأن النهاية الصغرى للنجاح هي ٥٠ وذلك بإظهار صندوق رسالة MsgBox به كلمة ناجح ؟



- ١ - بداية
- ٢ - أدخل درجة طالب في مادة اللغة العربية AR
- ٣ - إذا كان  $AR \geq 50$  إذن
- ٣ - ١ طباعة "الطالب ناجح"
- ٤ - نهاية

**الحل**

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر ( النتيجة ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```

Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim AR As Single
        AR = Me.TextBox1.Text
        If AR >= 50 Then
            MsgBox("الطالب ناجح")
        End If
    End Sub
End Class
  
```





- نلاحظ عند تشغيل البرنامج عدم ظهور صندوق الرسالة MsgBox عند إدخال أى قيمة أقل من ٥٠ لأنه سيكون ناتج الشرط  $AR \geq 50$  هو خطأ False ولن ينفذ أى شئ من جملة If ثم يتم تنفيذ الأوامر التي تلي End If وهو End Sub أى إنهاء الإجراء .
- يمكن كتابة الكود السابق كما يلي حيث ستكون جملة If فى سطر واحد وعندها لا يتم كتابة أمر End If كما يلي:

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim AR As Single
        AR = Me.TextBox1.Text
        If AR >= 50 Then MsgBox(" الطالب ناجح ")
    End Sub
End Class
```

### ثانياً : التفرع باستخدام جملة If ... Then ... Else ... :

وهي من أوامر التفرع لإختيار بديل من بين بديلين ( ثنائية الإختيار أو Double Selection ).  
وختلف عن الجملة السابقة فى وجود كود ينفذ عند تحقق الشرط True وكود آخر ينفذ عند عدم تحقق الشرط False

الصيغة العامة لهذه الجملة

If (التعبير الشرطى أو الشرط) Conditional Expression Then

Code1

Else

Code2

End If

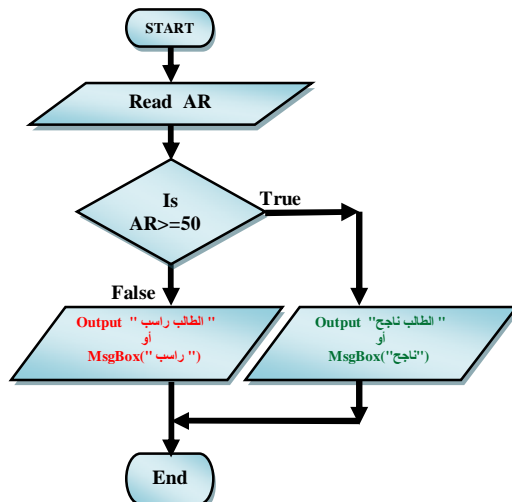
الأوامر التي ستنفذ فى حالة ما يكون  
ناتج الشرط صواب True

الأوامر التي ستنفذ فى حالة ما يكون  
ناتج الشرط خطأ False

ما يكتب بعد If هو الشرط أو التعبير الشرطى وفى حالة  
١ - ما يكون ناتج هذا الشرط صواب True يتم تنفيذ  
الأوامر Code1 التي تلي Then حتى الوصول إلى  
Else ثم يتم تنفيذ الأوامر التي تلي End If .  
٢ - وفى حالة ما يكون ناتج هذا الشرط خطأ False يتم  
تنفيذ الأوامر Code2 التي تلي Else حتى  
الوصول إلى End If ثم يتم تنفيذ الأوامر التي تلي  
End If .

### ٢ : تدريب :

قم بتحويل خطوات الحل و خريطة التدفق التالية كما درستها فى الفصل الدراسى الأول لعمل  
برنامج لعملية قراءة درجة طالب فى مادة اللغة العربية لمعرفة ما إذا كان ناجحاً أو راسباً علماً بأن  
النهاية الصغرى للنجاح هي ٥٠ وذلك بإظهار صندوق رسالة MsgBox به كلمة ناجح أو راسب ؟



١ - بداية

٢ - أدخل درجة طالب فى مادة اللغة العربية AR

٣ - إذا كان  $AR \geq 50$  إذن

٣ - ١ - طباعة " الطالب ناجح "

٤ - غير ذلك :

٤ - ١ - طباعة " الطالب راسب "

٤ - نهاية

## الحل

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر ( النتيجة ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim AR As Single
        AR = Me.TextBox1.Text
        If AR >= 50 Then
            MsgBox("الطالب ناجح")
        Else
            MsgBox("الطالب راسب")
        End If
    End Sub
End Class
```

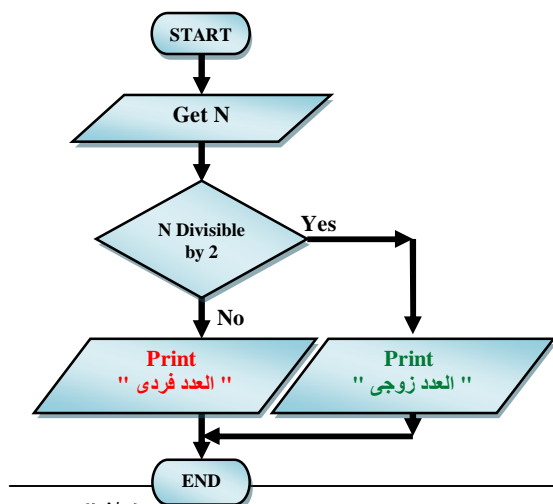


- كلمة Else تعني تنفيذ ما بعدها اذا كان ناتج الشرط خطأ ( False )
- اذا كان ناتج التعبير الشرطي صحيحاً ( True ) سيتم تنفيذ الأوامر التي تلي كلمة Then وستظهر رسالة بها كلمة "الطالب ناجح"
- أما اذا كان ناتج التعبير الشرطي خطأ ( False ) سيتم تنفيذ الأوامر التي تلي كلمة Else وستظهر رسالة بها كلمة "الطالب راسب"
- يمكن كتابة الكود Code السابق كما يلي حيث ستكون جملة If في سطر واحد وعندها لا يتم كتابة أمر End If كما يلي:

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim AR As Single
        AR = Me.TextBox1.Text
        If AR >= 50 Then MsgBox("الطالب ناجح") Else MsgBox("الطالب راسب")
    End Sub
End Class
```

### 1 تدريب ٣ :

قم بتحويل خطوات الحل و خريطة التدفق التالية كما درستها في الفصل الدراسي الأول لعمل برنامج لإدخال رقم ثم طباعة نوع العدد ( زوجي أو فردي ) ، وذلك بإظهار صندوق رسالة MsgBox به عبارة ( العدد زوجي ) أو ( العدد فردي ) ؟



١ - بداية

٢ - أدخل العدد N

٣ - إذا كان قيمة العدد N تقبل القسمة على ٢ بدون باقى إذن :

٣ - ١ - طباعة " العدد زوجي "

٤ - غير ذلك :

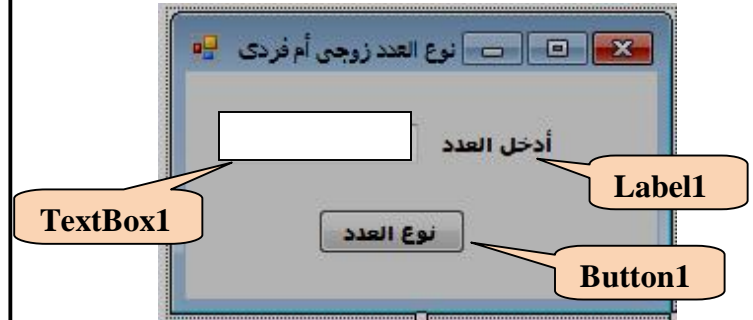
٤ - ١ - طباعة " العدد فردي "

٥ - نهاية

## الحل

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر ( نوع العدد ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N As Long
        N = Me.TextBox1.Text
        If N Mod 2 = 0 Then
            MsgBox("العدد زوجي")
        Else
            MsgBox("العدد فردي")
        End If
    End Sub
End Class
```



تم استخدام التعبير الشرطي  $N \text{ Mod } 2 = 0$  والذي يتكون من جزئين : الأول : التعبير الحسابي  $N \text{ Mod } 2$  والمعامل Mod هنا هو معامل حسابي يعطي باقى قسمة عددين . والثاني : هو مقارنة بين ناتج التعبير الحسابي السابق وبين القيمة 0 . فإذا كان باقى قسمة العدد N على العدد 2 يساوى 0 وهذا يعنى تحقق الشرط True يكون العدد زوجي ، وإذا كان غير ذلك False يكون العدد فردي

- يمكن كتابة الكود Code السابق كما يلي حيث ستكون جملة If في سطر واحد وعندها لا يتم كتابة أمر End If كما يلي:

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N As Long
        N = Me.TextBox1.Text
        If N Mod 2 = 0 Then MsgBox("العدد زوجي") Else MsgBox("العدد فردي")
    End Sub
End Class
```

### ثالثا : التفرع باستخدام جملة If ... Then ... ElseIf ... :

وهي من أوامر التفرع وهي تستخدم عند وجود أكثر من تعبير شرطي أو أكثر من إختيار أو بديل ( متعددة الإختيار أو Multiple Selection ) .

الصيغة العامة لهذه الجملة

```
If Conditional Expression Then
    Code1
ElseIf Conditional Expression Then
    Code2
ElseIf Conditional Expression Then
    Code3
.
.
Else
    Code7
End If
```

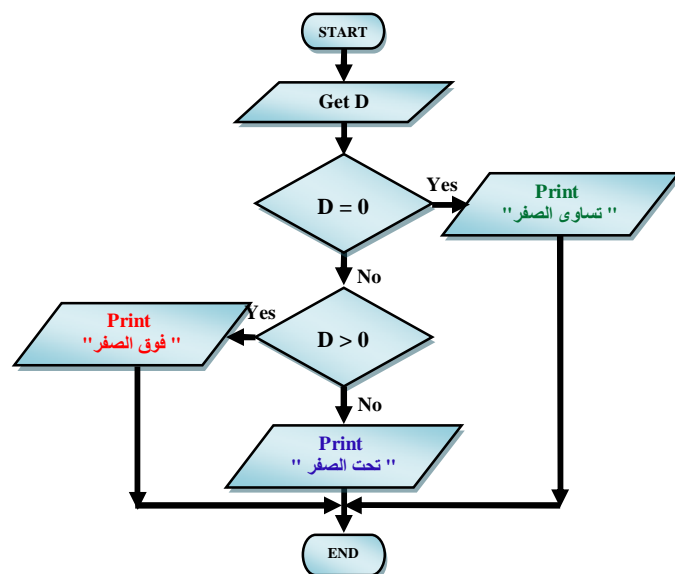
إذا تحقق أى شرط من الشروط فى جملة If أو أى من جمل ElseIf وكان ناتجه True نفذت الأوامر التى بعد Then الخاصة بهذا الشرط ، ولا يتم الانتقال إلى أى شرط آخر فى جملة If ولكن يتم الانتقال إلى جملة End If فى حالة :

- ١ - ما يكون ناتج الشرط لجملة If الأولى صواب True يتم تنفيذ الأوامر Code1 التى تلى Then ثم يتم الانتقال إلى جملة End If .
- ٢ - وفى حالة ما يكون ناتج هذا الشرط خطأ False يتم الانتقال إلى جملة ElseIf التالية فإذا كان ناتج الشرط لها صواب True يتم تنفيذ الأوامر Code2 التى تلى Then ثم يتم الانتقال إلى جملة End If ..
- ٣ - وهكذا ..... فإذا لم يتحقق أى من الشروط السابقة يتم تنفيذ الأوامر Code7 التى تلى Else الأخيرة ثم يتم الانتقال إلى جملة End If



## تدريب ٤ :

قم بتحويل خطوات الحل و خريطة التدفق التالية كما درستها في الفصل الدراسي الأول لعمل برنامج لإدخال درجة الحرارة في صندوق نص TextBox ، لمعرفة هل الدرجة ( فوق الصفر ) أو ( تحت الصفر ) أو ( تساوي الصفر ) و إظهارها في مربع عنوان Label ؟



- ١ - بداية
- ٢ - أدخل درجة الحرارة المنوية D
- ٣ - إذا كان  $D = 0$  إذن :
  - ٣ - ١ طباعة "تساوي الصفر"
  - ٣ - ٢ إذهب للخطوة رقم ٦
- ٤ - غير ذلك :
  - ٤ - ١ إذا كان  $D > 0$  إذن :
    - ٤ - ١ - ١ طباعة "فوق الصفر"
    - ٤ - ١ - ٢ إذهب للخطوة رقم ٦
  - ٤ - ٢ غير ذلك :
    - ٤ - ١ - ٥ طباعة "تحت الصفر"
- ٥ - غير ذلك :
- ٦ - نهاية

## الحل

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر ( إختبار ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```

Public Class Form1
    Private Sub Button1_Click(ByVal ...
        Dim Degree As Single
        Degree = Me.TextBox1.Text
        If Degree = 0 Then
            Me.Label2.Text = "درجة الحرارة تساوي الصفر"
        ElseIf Degree > 0 Then
            Me.Label2.Text = "درجة الحرارة فوق الصفر"
        Else
            Me.Label2.Text = "درجة الحرارة تحت الصفر"
        End If
    End Sub
End Class
  
```

TextBox1

Label2

Button1



- تم الإعلان عن المتغير Degree من النوع الرقمي الغير صحيح Single لإحتمال إدخال رقم عشري لدرجة الحرارة .
- تبدأ جملة If بالتعبير الشرطي  $\text{If Degree} = 0$  وفي حالة ما يكون ناتجه صحيح ( True ) ستظهر رسالة الدرجة تساوي صفر
- وإذا كان ناتجه خطأ يكون التعبير الشرطي الثاني هو  $\text{ElseIf Degree} > 0$  ، وفي حالة ما يكون ناتجه صحيح ( True ) ستظهر رسالة الدرجة أكبر من صفر
- وإذا كان ناتجه خطأ بالتالي يبقى الإحتمال الأخير هو  $\text{Degree} < 0$  لأنه هو الإحتمال الوحيد الباقي والمعبّر عنه ب Else لذلك لا يتم كتابته صريحاً في الكود .

- عند الانتقال إلى وضع اختبار البرنامج Start Debugging بعد الضغط على مفتاح F5 من لوحة المفاتيح ثم لم نقم بإدخال أي قيم بصندوق النص TextBox1. تظهر لنا رسالة الخطأ التالية من النوع Runtime Errors والتي توضح ( أنه لا يجوز تحويل القيمة النصية String الفارغة "" إلى نوع بيان رقمي صحيح Single ) لأن المترجم Compiler لم يستطع القيام بالتحويل الضمني من النوع الحرفي String إلى النوع الرقمي Single كما ذكرنا في الفصل الأول .

```
Dim Degree As Single
Degree = Me.TextBox1.Text
If Degree = 0 Then
    Me.Label2.Text = "درجة الحرارة تساوى الصفر"
ElseIf Degree > 0 Then
    Me.Label2.Text = "درجة الحرارة فوق الصفر"
ElseIf Degree < 0 Then
    Me.Label2.Text = "درجة الحرارة تحت الصفر"
End If
```

InvalidCastException was unhandled  
Conversion from string "" to type 'Single' is not valid.

- ويمكن منع حدوث هذا الخطأ بطريقتين :

**الطريقة الأولى :** يمكننا استخدام جملة Try Catch End Try التي درسناها في الفصل الأول كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal ... As Object)
        Dim Degree As Single
        Try
            Degree = Me.TextBox1.Text
            If Degree = 0 Then
                Me.Label2.Text = "درجة الحرارة تساوى الصفر"
            ElseIf Degree > 0 Then
                Me.Label2.Text = "درجة الحرارة فوق الصفر"
            Else
                Me.Label2.Text = "درجة الحرارة تحت الصفر"
            End If
        Catch ex As Exception
            MsgBox("أدخل عدد في صندوق النص")
            Me.TextBox1.Focus()
            Me.TextBox1.Text = ""
        End Try
    End Sub
End Class
```

- نلاحظ في هذا الكود عرض صندوق رسالة MsgBox يطالب المستخدم بإدخال عدد بدلاً من قيمة حرفية أو قيمة حرفية خالية في صندوق النص TextBox1.
- نلاحظ استخدام وسيلة هي Focus() خاصة بصندوق النص TextBox1 ، وهي تعني نقل التركيز ووضع مؤشر الكتابة بداخل صندوق النص TextBox1
- نلاحظ استخدام الكود Me.TextBox1.Text = "" وهو يعنى تخصيص القيمة "" أو السلسلة النصية الفارغة للخاصية Text لأداة صندوق النص TextBox1.

**الطريقة الثانية :** يمكننا استخدام جملة If إضافية ملحق بها الدالة IsNumeric() كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal ... As Object)
        Dim Degree As Single
        If IsNumeric(Me.TextBox1.Text) Then
            Degree = Me.TextBox1.Text
            If Degree = 0 Then
                Me.Label2.Text = "درجة الحرارة تساوى الصفر"
            ElseIf Degree > 0 Then
                Me.Label2.Text = "درجة الحرارة فوق الصفر"
            Else
                Me.Label2.Text = "درجة الحرارة تحت الصفر"
            End If
        Else
            MsgBox("أدخل عدد في صندوق النص")
            Me.TextBox1.Focus()
            Me.TextBox1.Text = ""
        End If
    End Sub
End Class
```

- نلاحظ استخدام جملة If إضافية If IsNumeric(Me.TextBox1.Text) لاختبار قيمة الخاصية Text لأداة صندوق النص TextBox1 هل هي قيمة عددية أم لا وفي حالة أن يكون الناتج True يتم تنفيذ الكود المطلوب ، أما في حالة أن يكون الناتج False أي أن القيمة غير عددية يتم تنفيذ الكود Code البديل الذي يلي Else ويكون من:
- ١ - عرض صندوق رسالة MsgBox يطالب المستخدم بإدخال عدد بدلاً من قيمة حرفية أو قيمة حرفية خالية في صندوق النص TextBox1
- ٢ - نقل التركيز أو نقل مؤشر الكتابة بداخل صندوق النص TextBox1 .
- ٣ - تخصيص القيمة "" أو السلسلة النصية الفارغة للخاصية Text لأداة صندوق النص

- الدالة ( IsNumeric ) : هو أحد الدوال المعرفة Predefined Functions التي تستخدم لمعرفة محتوى ( متغير أو ثابت أو محتوى صندوق نص TextBox أو عنصر في صندوق القائمة ListBox أو عنصر في صندوق التحرير والسرد ComboBox ) هل هو عددي أم غير عددي ، ويكون ناتج هذه الدالة True أو False .
- الأمر ( IsNumeric ) : لها مدخل أو وسيطة أو معامل Argument بين القوسين ولها ناتج منطقي هو True أو False .

**أمثلة على استخدام الدالة ( IsNumeric ) وناتجها حيث يتم وضع ناتج هذه الدالة في صندوق رسالة MsgBox :**

الناتج	الأمر
False حيث أن القيمة داخل القوسين غير عددية	MsgBox(IsNumeric("Hassan"))
True حيث أن القيمة داخل القوسين عددية	MsgBox(IsNumeric(10))
True حيث أن قيمة المتغير A عددية وهي 5	A = 5 MsgBox(IsNumeric(A))
False حيث أن قيمة المتغير B غير عددية وهي "Alhawy"	B = "Alhawy" MsgBox(IsNumeric(B))
True إذا كان صندوق النص TextBox1 يحتوي على قيمة عددية False إذا كان صندوق النص TextBox1 يحتوي على قيمة غير عددية	MsgBox(IsNumeric(TextBox1.Text))
True إذا كان العنصر المحدد في صندوق القائمة ListBox1 قيمة عددية False إذا كان العنصر المحدد في صندوق القائمة ListBox1 قيمة غير عددية	MsgBox(IsNumeric(ListBox1.SelectedItem))

**إستخدام المعاملات المنطقية Logical Operators مع الجملة الشرطية If :**

قد نحتاج إلى اختبار أكثر من تعبير شرطي معاً في نفس جملة If وفي هذه الحالة نربط بين التعبيرات الشرطية بالمعامل المنطقي And أو المعامل المنطقي Or .

المعامل المنطقي And يعطي ناتج نهائي True إذا كان ناتج كافة الشروط True

المعامل المنطقي And يعطي ناتج نهائي False إذا كان ناتج أحد الشروط False

المعامل المنطقي Or يعطي ناتج نهائي True إذا كان ناتج أحد الشروط True

المعامل المنطقي Or يعطي ناتج نهائي False إذا كان ناتج كافة الشروط False

**أمثلة على استخدام المعامل المنطقي And بفرض أن المتغير A والمتغير B متغيرات منطقية Boolean :**

قيمة المتغير B	قيمة المتغير A	التعبير الشرطي	الناتج
True	True	A And B	True
False	True	A And B	False
True	False	A And B	False
False	False	A And B	False

من الجدول السابق يتضح أن احتمال تحقق التعبير الشرطي A And B ( يكون ناتجه True ) هو احتمال وحيد عندما يكون ناتج كافة الشروط True

أمثلة على استخدام المعامل المنطقي Or بفرض أن المتغير A والمتغير B متغيرات منطقية Boolean :

قيمة المتغير B	قيمة المتغير A	التعبير الشرطي	النتيجة
True	True	A Or B	True
False	True	A Or B	True
True	False	A Or B	True
False	False	A Or B	False

من الجدول السابق يتضح أن احتمال تحقق التعبير الشرطي A Or B ( يكون ناتجه True ) هو ثلاث احتمالات عندما يكون ناتج أحد الشروط True

>> تدريب ٥ :

قم بإنشاء مشروع جديد يحتوي على نموذج Form التالي ، والمطلوب عند النقر على زر ( عرض النتيجة ) يعرض في Label4 الطالب ناجح أو الطالب راسب ، على أساس ناجح في حالة كلاً من درجتى التحريرى والعملية أكبر من أو تساوى ٥٠ ، وراسب إذا كانت أى من درجتى التحريرى والعملية أقل من ٥٠ ؟

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender ...
        Dim Degree1, Degree2 As Single
        If Isnumeric(TextBox1.Text) And Isnumeric(TextBox2.Text) Then
            Degree1 = Me.TextBox1.Text
            Degree2 = Me.TextBox2.Text

            If Degree1 >= 50 And Degree2 >= 50 Then
                Me.Label4.Text = "الطالب ناجح"
            Else
                Me.Label4.Text = "الطالب راسب"
            End If
        Else
            MsgBox("من فضلك أدخل أرقام في مربعات النص")
        End If
    End Sub
End Class
```

- من ملاحظة الكود السابق لكى يكون الطالب ناجحاً يجب أن كلاً من درجتى التحريرى Degree1 والعملية Degree2 أكبر من أو تساوى ٥٠ >= 50 ، وغير ذلك يعنى أن الطالب راسب
- إن استخدام المعامل المنطقي And فى التعبير الشرطي Degree1 >= 50 And Degree2 >= 50 أوجب تحقق جميع الشروط Degree1 >= 50 و Degree2 >= 50 ( أى True ) حتى يكون الناتج الكلى للتعبير الشرطي True.
- إن ما يتم كتابته بين If و Then هو تعبير شرطى سواء كان بسيط مثل Degree1 >= 50 أو مركب مثل Degree1 >= 50 And Degree2 >= 5 ، ولابد أن يكون ناتج هذا التعبير ناتج منطقى Boolean إما True أو False

قم بتعديل التدريب رقم ٥ السابق بحيث إذا تم تغيير شرط النجاح في حالة أي من الدرجتين التحريري أو العملي أكبر من أو تساوي ٥٠ ، وراسب إذا كانت كلاً من درجتى التحريري والعملي أقل من ٥٠ ؟

```
Public Class Form1
    Private Sub Button1_Click(ByVal sender ...
        Dim Degree1, Degree2 As Single
        If Isnumeric(TextBox1.Text) And Isnumeric(TextBox2.Text) Then
            Degree1 = Me.TextBox1.Text
            Degree2 = Me.TextBox2.Text

            If Degree1 >= 50 Or Degree2 >= 50 Then
                Me.Label4.Text = "الطالب ناجح"
            Else
                Me.Label4.Text = "الطالب راسب"
            End If
        Else
            MsgBox("من فضلك أدخل أرقام في مربعات النص")
        End If
    End Sub
End Class
```

TextBox1

TextBox2

Button1

- من ملاحظة الكود السابق لكي يكون الطالب ناجحاً يكفي أن تكون أي من درجتى التحريري Degree1 أو العملي Degree2 أكبر من أو تساوي ٥٠ ، وغير ذلك يعني أن الطالب راسب
- إن استخدام المعامل المنطقي Or في التعبير الشرطي Degree1 >= 50 Or Degree2 >= 50 لم يوجب تحقق جميع الشروط Degree1 >= 50 و Degree2 >= 50 ( أي True ) حتى يكون الناتج الكلي للتعبير الشرطي True. بل يكفي أن يتحقق ( أي True ) أحد الشروط حتى يكون الناتج الكلي للتعبير الشرطي True.

### التفرع باستخدام جملة : Select Case

وهي تشبه جملة If ... Then ... ElseIf ... غير أنها فعالة بشكل أكثر عندما يكون التفرع معتمداً على قيمة متغير Variable واحد فقط أو خاصية واحدة فقط ،  
وهي تجعل الكود Code مبسط ومفهوم بشكل أكثر من جملة If السابقة . وهي تستخدم في حالة وجود احتمال للتفرع أو أكثر

الصيغة العامة لهذه الجملة

```
Select Case variable
    Case value1
        Code1
    Case value2
        Code2
    Case value3
        Code3
    .
    .
    Case Else
        Code9
End Select
```

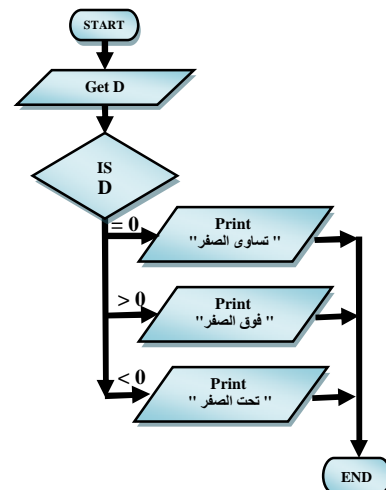
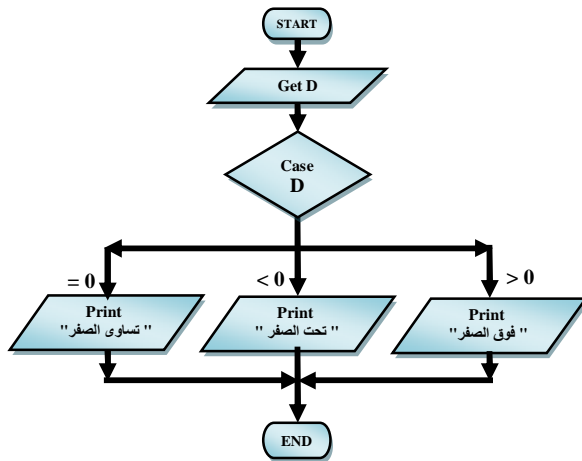
- تعتمد جملة Select Case في تفرعها على أكثر من قيمة ( Value1 و Value2 و Value3 و .... ) يعتمد غالباً على مقارنة هذه القيم بقيمة متغير وحيد variable أو قيمة خاصية واحدة ، حيث يتم الآتي :  
١ - في أمر Case value1 إذا كان قيمة المتغير variable في مجال أو حدود القيمة الأولى Value1 أي صواب True يتم تنفيذ الأوامر Code1 الأولى ثم يتم الانتقال إلى جملة End Select .  
٢ - وفي حالة ما يكون قيمة المتغير variable ليس في مجال أو حدود القيمة الأولى Value1 أي False يتم الانتقال إلى أمر Case value2 فإذا كان قيمة المتغير variable في مجال أو حدود القيمة الثانية Value2 أي صواب True يتم تنفيذ الأوامر Code2 الثانية ثم يتم الانتقال إلى جملة End Select .  
٣ - وهكذا ..... فإذا لم يتحقق أي من العمليات المنطقية السابقة (إذا لم تكن قيمة المتغير مساوية لأي من القيم السابقة ) حتماً يتم تنفيذ الأوامر Code9 التي تلي Case Else الأخيرة ثم يتم الانتقال إلى جملة End Select .



يمكن كتابة أمر Case Else الأخير بصورة أخرى مثل Case value# وعندها يتم كتابة قيمة أخرى value#، وإذا لم يتحقق أى من المقارنات (الأسئلة) السابقة لا يتم تنفيذ أى أمر Code ثم يتم الانتقال إلى جملة End Select.

1/2 تدريب ٧ :

قم بتعديل الكود Code في تدريب ٤ السابق باستخدام جملة Select Case بدلاً من جملة If Then Else! أو قم بتحويل خريطة التدفق التالية لعمل برنامج لإدخال درجة الحرارة في صندوق نص TextBox ثم إظهار في مربع عنوان Label لمعرفة هل الدرجة (فوق الصفر) أو (تحت الصفر) أو (تساوى الصفر) باستخدام جملة Select.... Case ؟



يتضح من الخريطين السابقتين إمكانية التفرع إلى أكثر من فرعين من خط الإجهاد الخارج من رمز اتخاذ القرار حيث يحتوى رمز اتخاذ القرار على سؤال عن قيمة المتغير D، وحسب قيمته نتفرع إلى ٣ أكواد Codes مختلفة، وعلى هذا يكون الكود كما يلي :

```

Public Class Form1
    Private Sub Button1_Click(ByVal...
        Dim Degree As Single
        Try
            Degree = Me.TextBox1.Text
            Select Case Degree
            Case 0
                Me.Label2.Text = "درجة الحرارة تساوى الصفر"
            Case > 0
                Me.Label2.Text = "درجة الحرارة فوق الصفر"
            Case < 0
                Me.Label2.Text = "درجة الحرارة تحت الصفر"
            End Select
            Catch ex As Exception
                MsgBox("أدخل عدد في صندوق النص")
                Me.TextBox1.Focus()
                Me.TextBox1.Text = ""
            End Try
        End Sub
    End Class
  
```

يمكن كتابة أمر Case < 0 بطريقة أخرى مثل Case Else وهذا يعنى إذا لم يتحقق أى من مجالات الأسئلة السابقة (تساوى الصفر أو أكبر من الصفر) حتماً يتم تنفيذ الكود Code الذى يلي Case Else ثم يتم الانتقال إلى جملة End Select.

قم بإنشاء مشروع جديد يحتوي على نموذج Form التالي ، يوضح شرح درس رياضيات للصف الثالث الابتدائي ، بحيث عند النقر على أي عنصر من عناصر الأداة ListBox1 يعرض عدد الأحرف Sides وعدد الأوجه Faces وعدد الرؤوس Vertices للمجسم المحدد ؟



```
Public Class Form1
```

```
Private Sub ListBox1_SelectedIndexChanged (By
```

```
Dim x As Byte
```

```
x = Me.ListBox1.SelectedIndex
```

```
Select Case x
```

```
Case 0
```

```
Me.Label5.Text = 12
```

```
Me.Label6.Text = 6
```

```
Me.Label7.Text = 8
```

```
Case 1
```

```
Me.Label5.Text = 12
```

```
Me.Label6.Text = 6
```

```
Me.Label7.Text = 8
```

```
Case 2
```

```
Me.Label5.Text = 6
```

```
Me.Label6.Text = 3
```

```
Me.Label7.Text = 4
```

```
Case 3
```

```
Me.Label5.Text = "لا يوجد"
```

```
Me.Label6.Text = "لا يوجد"
```

```
Me.Label7.Text = "لا يوجد"
```

```
End Select
```

```
End Sub
```

```
End Class
```

تعريف متغير اسمه x من نوع بيان عددي صحيح Byte

تخصيص قيمة فهرس العنصر المحدد لأداة ListBox1 في المتغير x

الكود في حالة اختيار العنصر ترتيبه ( فهرسه ) 0 أي المكعب

الكود في حالة اختيار العنصر ترتيبه ( فهرسه ) 1 أي متوازي المستطيلات

الكود في حالة اختيار العنصر ترتيبه ( فهرسه ) 2 أي الهرم الثلاثي

الكود في حالة اختيار العنصر ترتيبه ( فهرسه ) 3 أي الأسطوانة

- تم إضافة عناصر Items الأداة ListBox1 في نمط التصميم من خلال الخاصية Items من نافذة الخصائص .
- كل عنصر يتم إضافته للأداة ListBox1 له ترتيب أو فهرس Index ، بحيث يكون العنصر الأول ترتيبه 0 والثاني 1 .... وهكذا .
- وبالتالي يتم معرفة ترتيب أو فهرس Index العنصر المحدد SelectedIndex من خلال الخاصية SelectedIndex للأداة ListBox1 .
- وبالتالي عندما تكون قيمة الخاصية SelectedIndex هي 1 فهذا يعني تحديد العنصر الثاني .
- تم كتابة الكود السابق في إجراء الحدث SelectedIndexChanged الذي يعني عند حدوث تغير الفهرس Index المختار ، أي ترتيب أو فهرس العنصر المحدد ، وهو الحدث الافتراضي للأداة ListBox1 ويتحقق هذا الحدث عند تغير اختيار عنصر من عناصر الأداة ListBox1 .
- الملاحظات الخمس السابقة تنطبق أيضاً وبالكامل على الأداة ComboBox .

تدريب ٩ :

قم بتعديل الكود Code في تدريب ٨ السابق باستخدام جملة If ... Then ... ElseIf ... بدلاً من جملة Select Case ؟

```
Public Class Form1
    Private Sub ListBox1_SelectedIndexChanged(ByVal x As Byte)
        x = Me.ListBox1.SelectedIndex
        If x = 0 Then
            Me.Label5.Text = 12
            Me.Label6.Text = 6
            Me.Label7.Text = 8
        ElseIf x = 1 Then
            Me.Label5.Text = 12
            Me.Label6.Text = 6
            Me.Label7.Text = 8
        ElseIf x = 2 Then
            Me.Label5.Text = 6
            Me.Label6.Text = 3
            Me.Label7.Text = 4
        ElseIf x = 3 Then
            Me.Label5.Text = "لا يوجد"
            Me.Label6.Text = "لا يوجد"
            Me.Label7.Text = "لا يوجد"
        End If
    End Sub
End Class
```

A تدريب ١٠ : من دراستك لمادة العلوم تعلمت أن لكل عنصر رمز وعدد ذري وعدد كتلي ، قم بإنشاء مشروع يحتوي على نموذج Form التالي ، والمطلوب عند اختيار أحد العناصر ثم النقر على زر ( عرض ) يظهر رمز العنصر وعدده الذري والعدد الكتلي لهذا العنصر؟

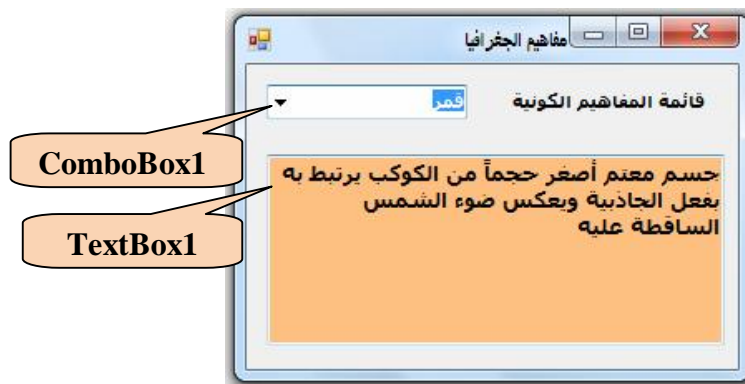
```
Private Sub Button1_Click(ByVal sender As ...
    If Me.RadioButton1.Checked = True Then
        Me.Label4.Text = "O"
        Me.Label5.Text = "8"
        Me.Label6.Text = "16"
    ElseIf Me.RadioButton2.Checked = True Then
        Me.Label4.Text = "Na"
        Me.Label5.Text = "11"
        Me.Label6.Text = "23"
    ElseIf Me.RadioButton3.Checked = True Then
        Me.Label4.Text = "Mg"
        Me.Label5.Text = "12"
        Me.Label6.Text = "24"
    ElseIf Me.RadioButton4.Checked = True Then
        Me.Label4.Text = "Al"
        Me.Label5.Text = "13"
        Me.Label6.Text = "27"
    Else
        MsgBox("يرجى اختيار أحد العناصر")
    End If
End Sub
```



- في هذا التدريب استخدمنا أدوات التحكم RadioButton لعرض العناصر الكيميائية بدلاً من ListBox ، وفي هذه الحالة اضطررنا لاستخدام جملة If ... Then ... ElseIf بدلاً من جملة Select Case .
- إن التفرع باستخدام جملة Select Case فعالة بشكل أكثر عندما يكون التفرع معتمداً على قيمة متغير Variable واحد فقط أو خاصية واحدة فقط .
- وبالتالي هناك صعوبة في استخدامها عند اختبار شرط يعتمد في تفرعه على أكثر من متغير أو خاصية . وهذا هو ما حدث في التدريب السابق ،
- في الملاحظة السابقة نستنتج أنه هناك صعوبة في تحويل الكود Code المكتوب بجملة If ... Then ... ElseIf إلى كود Code مكتوب بجملة Select Case . إلا إذا كان التفرع معتمداً على ( الشرط ) قيمة متغير Variable واحد فقط أو خاصية أداة واحدة فقط .
- من السهل تحويل الكود Code المكتوب بجملة Select Case التي تعتمد في صياغة المقارنة على قيمة متغير واحد فقط أو خاصية واحدة فقط إلى كود Code مكتوب بجملة If ... Then ... ElseIf لأن جملة If من الممكن أن تعتمد في صياغة الشرط على قيمة متغير أو أكثر أو قيمة خاصية أو أكثر .

### A تدريب ١١ :

من دراستك لمادة الجغرافيا تعلمت الكثير من المفاهيم مثل الجرة والكوكب والقمر والنجم و النيزك والشهاب ، قم بإنشاء مشروع يحتوي على النموذج التالي ، المطلوب هو عند اختيار أحد عناصر الأداة ComboBox يعرض في مربع نص TextBox تعريفه ؟



```
Public Class Form1
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
Select Case Me.ComboBox1.SelectedIndex
    Case 0
        Me.TextBox1.Text = " تجمعات كبيرة من النجوم المختلفة الأحجام والأشكال والأنواع "
    Case 1
        Me.TextBox1.Text = " جسم معتم لا يشع ضوءاً ولا حرارة ويعتمد ضوءه وحرارته من نجم قريب منه "
    Case 2
        Me.TextBox1.Text = " جسم معتم أصغر حجماً من الكوكب يرتبط به بفعل الجاذبية ويعكس ضوء الشمس الساقطة عليه "
    Case 3
        Me.TextBox1.Text = " جسم مضئ ملتهب يشع ضوءاً وحرارة "
    Case 4
        Me.TextBox1.Text = " جسم صلب كبير يحترق جزئياً عند احتكاكه بالغلاف الجوي "
    Case 5
        Me.TextBox1.Text = " جسم سماوي صلب يسبح في الفضاء حول الشمس ويجرى في السماء على هيئة سهام ضوئية "
    Case Else
        MsgBox(" يرجى اختيار أحد المفاهيم ")
    End Select
End Sub
End Class
```

- في هذا التدريب استخدمنا أداة التحكم ComboBox لعرض عناصر ( مفاهيم ) الجغرافيا وتم إضافة هذه العناصر من خلال الخاصية Items من نافذة الخصائص في نمط التصميم. كما استخدمنا الحدث SelectedIndexChanged والذي شرحناه في تدريب ٨
- لجعل مربع النص TextBox1 يعرض المعلومات على أكثر من سطر نغير قيمة الخاصية MultiLine إلى True من نافذة الخصائص

**تدريب ١٢ : قم بتعديل الكود Code في تدريب ١١ السابق باستخدام جملة If .. Then ... ElseIf بدلاً من جملة Case Select ؟**

```
Public Class Form1
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
    If Me.ComboBox1.SelectedIndex = 0 Then
        Me.TextBox1.Text = "تجمعات كبيرة من النجوم المختلفة الأحجام والأشكال والأنواع"
    ElseIf Me.ComboBox1.SelectedIndex = 1 Then
        Me.TextBox1.Text = "جسم معتم لا يشع ضوءاً ولا حرارة ويستمد ضوءه وحرارته من نجم قريب منه"
    ElseIf Me.ComboBox1.SelectedIndex = 2 Then
        Me.TextBox1.Text = "جسم معتم أصغر حجماً من الكوكب يرتبط به بفعل الجاذبية ويعكس ضوء الشمس الساقطة عليه"
    ElseIf Me.ComboBox1.SelectedIndex = 3 Then
        Me.TextBox1.Text = "جسم مضئ ملتهب يشع ضوءاً وحرارة"
    ElseIf Me.ComboBox1.SelectedIndex = 4 Then
        Me.TextBox1.Text = "جسم صلب كبير يحترق جزئياً عند احتكاكه بالغلاف الجوي"
    ElseIf Me.ComboBox1.SelectedIndex = 5 Then
        Me.TextBox1.Text = "جسم سماوي صلب يسبح في الفضاء حول الشمس ويجرى في السماء على هيئة سهام ضوئية"
    Else
        MsgBox("يرجى اختيار أحد المفاهيم")
    End If
End Sub
End Class
```

- في التدريب ١١ و ١٢ استخدمنا الأمر Case في جملة Case Else و الأمر Else في جملة If هما أمران متماثلان من حيث استخدامهما ففي حالة عدم تحقق أى من الشروط السابقة يتم حتماً تنفيذ الكود Code الذي يليها ، وفي حالة مثالنا هذا يظهر صندوق رسالة MsgBox يطلب من المستخدم اختيار أحد المفاهيم.

**1 تدريب ١٣ : قم بتعديل الكود Code في تدريب ٨ السابق باستخدام الخاصية SelectedItems لأداة**

```
Public Class Form1
    Private Sub ListBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
    Dim x As String
    x = Me.ListBox1.SelectedItem
    Select Case x
        Case "المكعب"
            Me.Label5.Text = 12
            Me.Label6.Text = 6
            Me.Label7.Text = 8
        Case "متوازي المستطيلات"
            Me.Label5.Text = 12
            Me.Label6.Text = 6
            Me.Label7.Text = 8
        Case "الهرم الثلاثي"
            Me.Label5.Text = 6
            Me.Label6.Text = 3
            Me.Label7.Text = 4
        Case "الأسطوانة"
            Me.Label5.Text = "لا يوجد"
            Me.Label6.Text = "لا يوجد"
            Me.Label7.Text = "لا يوجد"
    End Select
End Sub
```

تعريف متغير اسمه x من نوع بيان حرفي Sting

تخصيص قيمة العنصر المحدد لأداة ListBox1 في المتغير x

الكود في حالة اختيار العنصر المسمى المكعب

الكود في حالة اختيار العنصر المسمى متوازي المستطيلات

الكود في حالة اختيار العنصر المسمى الهرم الثلاثي

الكود في حالة اختيار العنصر المسمى الأسطوانة

في هذا التدريب استخدمنا الخاصية SelectedItems للأداة ListBox1 وهذه الخاصية لها قيمة من نوع بيان حرفي String لذا استخدمنا متغير في تخزينها هو المتغير x الذي تم الإعلان عنه بأنه متغير حرفي String.

° تدريب ١٤ : قم بتعديل الكود Code في تدريب ١١ السابق باستخدام الخاصية SelectedItems للأداة ComboBox1 :

```
Public Class Form1
    Private Sub ComboBox1_SelectedIndexChanged(ByVal sender As System.Object, ByVal e As
        Select Case Me.ComboBox1.SelectedItems
            Case "مجرة"
                Me.TextBox1.Text = "تجمعات كبيرة من النجوم المختلفة الأحجام والأشكال والأنواع"
            Case "كوكب"
                Me.TextBox1.Text = "جسم معتم لا يشع ضوءاً ولا حرارة ويستمد ضوءه وحرارته من نجم قريب منه"
            Case "قمر"
                Me.TextBox1.Text = "جسم معتم أصغر حجماً من الكوكب يرتبط به بفعل الجاذبية ويعكس ضوء الشمس الساقطة عليه"
            Case "نجم"
                Me.TextBox1.Text = "جسم مضيئ ملتهب يشع ضوءاً وحرارة"
            Case "نيزك"
                Me.TextBox1.Text = "جسم صلب كبير يحترق جزئياً عند احتكاكه بالغلاف الجوي"
            Case "شهاب"
                Me.TextBox1.Text = "جسم سماوي صلب يسبح في الفضاء حول الشمس ويجرى في السماء على هيئة سهام ضوئية"
            Case Else
                MsgBox("يرجى اختيار أحد المفاهيم")
        End Select
    End Sub
End Class
```

## الفصل الثالث ( الحلقات التكرارية والمؤقتات Loops and Timers )

الحلقات التكرارية : تستخدم لتكرار كود معين عدد من المرات .

أولاً : استخدام الجملة For ..... Next :

يستخدم لتكرار تنفيذ كود Code لعدد محدد من المرات في حالة معرفة عدد مرات التكرار مبسقا .

الصيغة العامة لهذه الجملة

For Variable = Start To End

Code

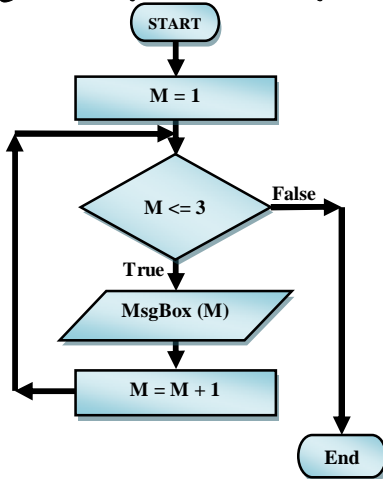
الأوامر التي سيتم تكرارها

Next

يبدأ الإعلان عن الحلقة التكرارية بكلمة For وتنتهي بكلمة Next ويستمر التكرار طالما قيمة العداد لم تتخطى قيمة نهاية العداد End ويحدد مع For :

- ١ - اسم متغير Variable يطلق عليه متغير عداد Counter الحلقة نوع بياناته عددي ( صحيح أو غير صحيح ) .
- ٢ - والعداد له قيمة بداية Start عددية (صحيح أو غير صحيح) .
- ٣ - والعداد له قيمة نهاية End عددية ( صحيح أو غير صحيح ) .
- ٤ - الكود Code المراد تكراره لعدد محدد من المرات .

٠ تدريب ١ : قم بتحويل خطوات الحل و خريطة التدفق التالية كما درستها في الفصل الدراسي الأول لعمل برنامج لعرض الأعداد من ١ إلى ٣ من خلال إظهار صندوق رسالة MsgBox وذلك عند النقر على زر Button ؟



- ١ - بداية
- ٢ - M = 1
- ٣ - إذا كان M <= 3 إذن :
  - ٣ - ١ طباعة قيمة M
  - ٣ - ٢ M = M + 1
  - ٣ - ٣ اذهب للخطوة رقم ٣
- ٤ - غير ذلك :
- ٤ - ١ نهاية

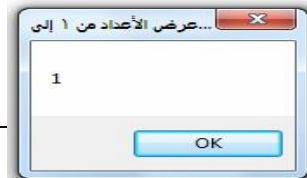
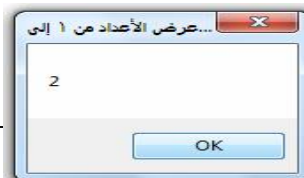
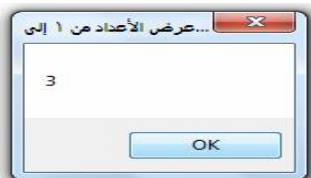
يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر ( عرض الأعداد من ١ إلى ٣ ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```

Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim M As Byte
        For M = 1 To 3
            MsgBox(M)
        Next
    End Sub
End Class
    
```



نلاحظ عند تشغيل البرنامج ثم نقر زر الأمر ( عرض الأعداد من ١ إلى ٣ ) ظهور ٣ صناديق رسالة MsgBox تباعاً ولن يظهر صندوق الرسالة الثاني إلا بعد النقر على زر OK في صندوق الرسالة الأول ..... وهكذا.



يمكن تتبع سير الكود Code كما هو موضح بالجدول التالي :

الأمـر Code	الوصف	قيمة المتغير M	ناتج صندوق الرسالة
Dim M As Byte	تعريف المتغير M من نوع عددي صحيح Byte	0	—
For M = 1 To 3	بداية الإعلان عن جملة الحلقة For عدادها المتغير M بقيمة بداية هي ١ وقيمة نهاية هي ٣	1	—
MsgBox(M)	إظهار مربع رسالة بقيمة المتغير M	1	1
Next	نهاية جملة الحلقة For والتي تقوم : ١ - بزيادة عداد الحلقة M بواحد M=M+1 ٢ - هل قيمة M أكبر من قيمة النهاية ٣ (False)	2	—
MsgBox(M)	إظهار مربع رسالة بقيمة المتغير M	2	2
Next	نهاية جملة الحلقة For والتي تقوم : ١ - بزيادة عداد الحلقة M بواحد M=M+1 ٢ - هل قيمة M أكبر من قيمة النهاية ٣ (False)	3	—
MsgBox(M)	إظهار مربع رسالة بقيمة المتغير M	3	3
Next	نهاية جملة الحلقة For والتي تقوم : ١ - بزيادة عداد الحلقة M بواحد M=M+1 ٢ - هل قيمة M أكبر من قيمة النهاية ٣ (True)	4	—

• من تتبع الكود Code في الجدول السابق نجد أن وظيفة الأمر Next : كما وردت بالكتاب المدرسي هي :

١ - زيادة عداد الحلقة ( المتغير M ) بقيمة معدل الزيادة أي ( بواحد افتراضياً )  $M=M+1$  في كل مرة.

٢ - مقارنة القيمة الجديدة لعداد الحلقة ( المتغير M ) بقيمة النهاية End للحلقة التكرارية ( العدد ٣ )

وفي حالة :

أ - عدم خطيها ( نقصانها عن ) قيمة النهاية يتم تكرار الكود Code أو الأوامر التي تلي الأمر For

ب - خطيها ( زيادتها عن ) قيمة النهاية يتم الخروج من الحلقة وتنفيذ الكود Code الذي يلي الأمر Next وهو

End Sub

• تم تحديد نوع بيان Data Type للمتغير M من نوع عددي صحيح Byte وليس Integer وفقاً للاستخدام

الأمثل للذاكرة

٢ : قم بتعديل تدريب ١ السابق بحيث يعرض الأعداد في صندوق نص TextBox ؟

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر الأمر ( عرض الأعداد من ١ إلى ٣ ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal .....
        Dim M As Byte
        Me.TextBox1.Text = ""
        For M = 1 To 3
            Me.TextBox1.Text = Me.TextBox1.Text & M
        Next
    End Sub
End Class
```



• من تتبع الكود Code السابق تجد :

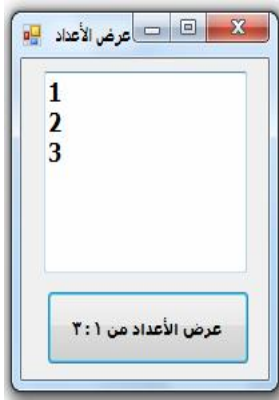
١ - الأمر `Me.TextBox1.Text = Me.TextBox1.Text & M` المقصود به وضع ( إضافة ) قيمة المتغير `M` بجانب ما هو موجود بداخل صندوق النص `TextBox1` وتم ذلك بكتابة معامل الربط `&` والذي يستخدم في وصل ( ربط ) سلسلتين حرفيتين ببعضهما.

٢ - في حالة كتابة الأمر السابق بهذه الطريقة `Me.TextBox1.Text = M` لن نلاحظ تغير قيم المتغير `M` أثناء التنفيذ ، وإنما سيظهر آخر قيمة فقط له وهي 3 كما يلي  لأننا لم نضف قيمة المتغير للمحتوى الموجود بداخل صندوق النص `TextBox1`

٣ - عند النقر `Click` على زر الأمر ( عرض الأعداد من ١ إلى ٣ ) أكثر من مرة سيظهر صندوق النص به الناتج  ولتفادي ذلك يتم إضافة الأمر `Me.TextBox1.Text = ""` قبل الحلقة التكرارية لمسح محتويات صندوق النص `TextBox1` ، أما إذا لم نستخدم هذا الأمر ونقرنا `Click` على زر الأمر (عرض الأعداد من ١

إلى ٣) ثلاث مرات مثلاً سيظهر صندوق النص به الناتج 123123123 كما يلي

٤ - كما يمكننا عرض الأرقام بداخل صندوق النص `TextBox1` بحيث يكون كل رقم في سطر جديد وذلك باستخدام الكلمة المحجوزة `vbCrLf` والتي تستخدم في إنشاء سطر جديد وهي تشير لرمز مفتاح الإدخال `Enter` من لوحة المفاتيح ، مع تغيير قيمة الخاصية `MultiLine` للأداة `TextBox1` لكي تكون `True` مع تغيير حجمها في وضع التصميم كما في الشكل التالي :

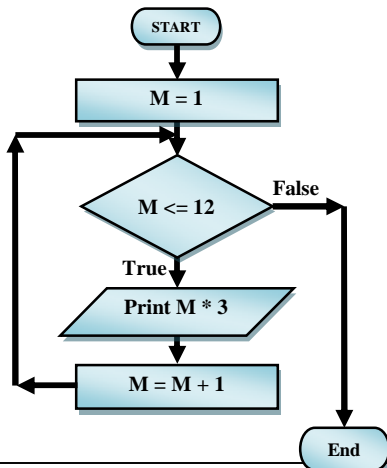


`Me.TextBox1.Text = Me.TextBox1.Text & M & vbCrLf` ←

الثابت الحرفي `vbCrLf` هي اختصار لـ `Visual Basic Carriage Return Line Feed`

### ١ تدريب ٣ :

قم بتحويل خطوات الحل وخريطة التدفق التالية كما درستها في الفصل الدراسي الأول لعمل برنامج لطباعة جدول ضرب العدد ٣ في صندوق نص `TextBox` وذلك عند النقر `Click` على زر `Button` ؟



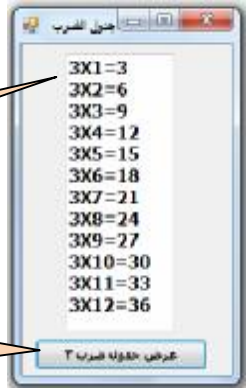
- ١ - بداية
- ٢ - `M = 1`
- ٣ - إذا كان `M <= 12` إذن :
  - ١ - ٣ - طباعة قيمة `M * 3`
  - ٢ - ٣ - `M = M + 1`
  - ٣ - ٣ - إذهب للخطوة رقم ٣
- ٤ - غير ذلك :
- ٤ - ١ - نهاية

يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود الـ Code البرمجة الخاص بزر ( عرض جدول ضرب ٣ ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal .....
        Dim M, prodect As Integer
        Dim str As String
        For M = 1 To 12
            str = 3 & "I " & M & "="
            prodect = 3 * M
            Me.TextBox1.Text = Me.TextBox1.Text & str & prodect & vbCrLf
        Next
    End Sub
End Class
```

TextBox1

Button1



• من تتبع الكود السابق نجد :

ز تم الإعلان عن متغير باسم product لتخزين حاصل الضرب به مع كل تغير في قيمة المتغير M (  $prodect = 3 * M$  )  
 k تم الإعلان عن متغير باسم str لتخزين شكل جملة حاصل الضرب (  $3I 1=$  أو  $3I 2=$  أو  $3I 3=$  أو ..... وهكذا )  
 كسلسلة نصية مع كل تغير في قيمة المتغير أو العدد M (  $str = 3 & "I " & M & "="$  )، حيث يتم تمثيل الأعداد ١ ، ٢ ، ٣ ، ٤ ، ..... ، ١٢ بالمتغير M  
 l تم وصل (ربط) محتوى صندوق النص TextBox1 بقيمة المتغير str وربطهم بقيمة المتغير product باستخدام معامل الربط &  
 m تم استخدام الثابت الحرفي vbCrLf لإضافة سطر جديد  
 n يمكن كتابة الكود السابق بدون استخدام هذه المتغيرات Variables والإكتفاء بالمتغير M حتى فصل على الأعداد ١ ، ٢ ، ٣ ، ٤ ، ..... ، ١٢ . كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal .....
        Dim M As Integer
        For M = 1 To 12
            Me.TextBox1.Text = Me.TextBox1.Text & 3 & "I " & M & "=" & 3 * M & vbCrLf
        Next
    End Sub
End Class
```

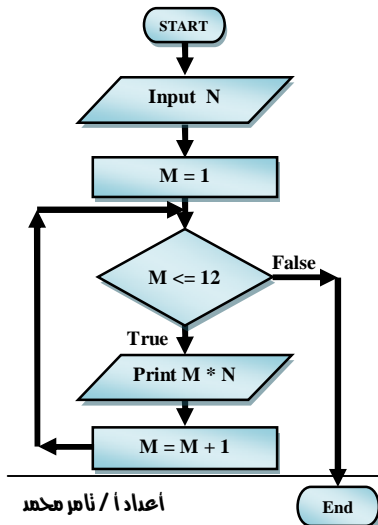
str

product

○ إذن السؤال الذي يتبادر إلى الذهن : لماذا إذن استخدام المتغيرات ؟ وذلك لتبسيط وتسهيل تتبع كود الـ Code البرنامج ،

○ تدريب ٤ :

قم بتطوير البرنامج بالتدريب ٣ السابق لطباعة جدول الضرب لأي رقم يتم ادخاله من خلال صندوق نص TextBox ؟ وعلى ذلك يكون خريطة التدفق وخطوات الحل له كما يلي :



١ - بداية

٢ - أدخل قيمة العدد N المراد عمل جدول الضرب له

٣ -  $M = 1$

٤ - إذا كان  $M \leq 12$  إذن :

٤ - ١ - طباعة قيمة  $M * N$

٤ - ٢ -  $M = M + 1$

٤ - ٣ - اذهب للخطوة رقم ٤

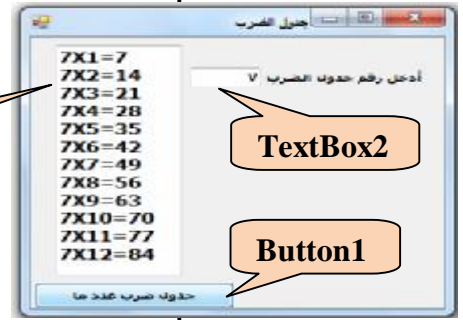
٥ - غير ذلك :

٥ - ١ - نهاية



يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود البرمجة الخاص بزر ( جدول ضرب عدد ما ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal .....
        Dim M, prodect , NUM As Integer
        Dim str As String
    j   NUM = Me.TextBox2.Text
        Me.TextBox1.Text = ""
    k   For M = 1 To 12
    k       str = NUM & "I " & M & "="
        prodect = NUM * M
        Me.TextBox1.Text = Me.TextBox1.Text & str & prodect & vbCrLf
    Next
    End Sub
End Class
```



• من تتبع الكود Code السابق نجد :

ز تم إضافة السطر NUM = Me.TextBox2.Text والهدف منه تخصيص قيمة الخاصية Text لأداة صندوق النص TextBox2 وتخزينه في المتغير العددي الصحيح NUM .  
k تم استبدال العدد ٣ في التدریب رقم ٣ بالمتغير NUM والذي يحتوي على قيمة الخاصية Text من مربع النص TextBox2

#### التحكم في البداية والنهاية ومقدار الزيادة في جملة For ..... Next :

في التدريبات السابقة كانت قيمة البداية للعداد Counter أو المتغير Variable دائماً أصغر من قيمة النهاية له ، ومقدار زيادته دائماً تساوى ١ . حيث يمكنك إضافة كلمة Step في نهاية الأمر والتي تعنى الخطوة أو مقدار الزيادة ، حيث يمكن :  
ز جعل قيمة الزيادة أكبر من الواحد مثل ٢ أو ٣ أو ... إلخ أو رقم غير صحيح ( عشري ) مثل ٢,٥ أو ٥ أو ٤,٥ وهكذا ،  
k كما يمكنك جعل قيمة البداية Start أكبر من قيمة النهاية End وفي هذه الحالة يجب أن تكون قيمة الزيادة Step سالبة  
l كما يمكن أن تكون قيمة البداية Start و قيمة النهاية End و قيمة الزيادة Step رقم صحيح أو عشري أو متغير عددي أو ثابت عددي ، رقم سالب أو موجب .

الصيغة العامة لهذه الجملة

For Variable = Start To End [Step n]

Code

الأوامر التي سيتم تكرارها

Next [Variable]

يبدأ الإعلان عن الحلقة التكرارية بكلمة For وتنتهى بكلمة Next ويحدد مع For :

- ١ - إسم متغير Variable يطلق عليه متغير عداد Counter الحلقة نوع بيانه عددي ( صحيح أو غير صحيح ) .
- ٢ - والعداد له قيمة بداية Start عددية ( صحيح أو غير صحيح )
- ٣ - والعداد له قيمة نهاية End عددية ( صحيح أو غير صحيح ) .
- ٤ - قيمة الزيادة Step وهي اختيارية أى يمكن عدم كتابتها وفي هذه الحالة تأخذ قيمة افتراضية هي ١
- ٥ - الكود Code المراد تكراره لعدد محدد من المرات .
- ٦ - Next أى نهاية الحلقة التكرارية ويتبعها إسم المتغير و إسم [المتغير] إختيارى يمكن عدم كتابته وفي هذه الحالة يأخذ قيمة افتراضية هي إسم المتغير الذى يلي كلمة For.

من الهيكل العام السابق تم وضع بعض الكلمات بين قوسين بهذا الشكل [ ] وهذا للدلالة على أنها إختيارية أى يمكن عدم كتابتها وفي هذه الحالة تأخذ قيمة افتراضية كما شرحنا في المستطيل السابق .



## أمثلة على استخدام جملة For :

م	المثال	الكود
١	لعرض الأعداد الفردية من ١ إلى ١٠	<pre>Dim I As Byte For I = 1 To 10 Step 2     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٢	لعرض الأعداد الزوجية من ٢ إلى ١٠	<pre>Dim I As Byte For I = 2 To 10 Step 2     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٣	لعرض الأعداد التي تقبل القسمة على ٣ أو مضاعفات رقم ٣ من ٣ إلى ١٠	<pre>Dim I As Byte For I = 3 To 10 Step 3     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٤	لعرض الأعداد الفردية مرتبة تنازلياً من ٩ إلى ١	<pre>Dim I As Byte For I = 9 To 1 Step -2     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٥	لعرض الأعداد الزوجية مرتبة تنازلياً من ١٠ إلى ٢	<pre>Dim I As Byte For I = 10 To 2 Step -2     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٦	لعرض الأعداد من ١.٥ إلى ٠.٥ بتناقص قدره ٠.٥ كل مرة	<pre>Dim I As Single For I = 1.5 To 0.5 Step -0.5     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٧	لعرض الأعداد من ١ إلى قيمة B بمعدل زيادة قيمته C	<pre>Dim I, B, C As Single For I = 1 To B Step C     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>
٨	لعرض الأعداد من قيمة A إلى قيمة B بمعدل زيادة أو نقصان قيمته C	<pre>Dim I, A, B, C As Single For I = A To B Step C     Me.TextBox1.Text = Me.TextBox1.Text &amp; I &amp; vbCrLf Next</pre>

• من الجدول السابق نجد أن :

z يمكن تحديد معدل ( قيمة ) الزيادة للمتغير أو العداد Counter بكلمة Step ثم كتابة الرقم .

k معدل ( قيمة ) الزيادة للمتغير أو العداد Counter يجب أن يكون سالباً في حالة أن تكون قيمة البداية أكبر من قيمة النهاية .

l يمكن لقيمة البداية أو النهاية أو معدل الزيادة أن يكون عدد عشري وفي هذه الحالة يجب تعريف متغير الحلقة من نوع

بيان Data Type يقبل الكسور العشرية مثل النوع Single

m يمكن لقيمة البداية أو النهاية أو معدل الزيادة أن يكون متغير Variable أو ثابت عددي .

n يجب أن يكون عداد الحلقة Counter متغير Variable عددي مناسب .

» تدريب ٥ : قم بإنشاء مشروع لعرض الأعداد التي تقبل القسمة على ٥ أو مضاعفات العدد ٥ من ١٠٥ إلى ٤٠٠٠ في صندوق نص TextBox وذلك عند النقر على زر Button ؟

```
Public Class Form1
Private Sub Button1_Click(ByVal .....
Dim M As Short
Me.TextBox1.Text = ""
For M = 105 To 4000 Step 5
Me.TextBox1.Text = Me.TextBox1.Text & M & vbCrLf
Next
End Sub
End Class
```

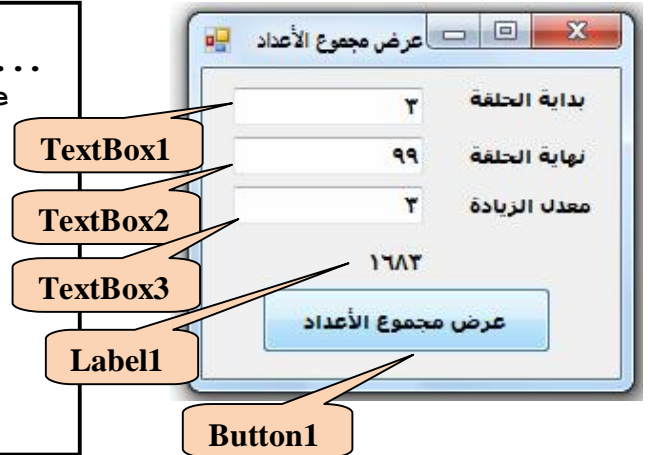


¼ تدريب ٦ : قم بكتابة كود البرمجة اللازم لعرض مجموع الأعداد التي تقبل القسمة على ٥ أو مضاعفات العدد ٥ من ١٠٥ إلى ٤٠٠٠ في صندوق نص TextBox وذلك عند النقر على زر Button ؟

```
Public Class Form1
Private Sub Button1_Click(ByVal .....
Dim M , Sum As Integer
Me.TextBox1.Text = ""
For M = 105 To 4000 Step 5
Sum = Sum + M
Next
Me.TextBox1.Text = Sum
End Sub
End Class
```

½ تدريب ٧ : قم بتعديل الكود السابق في تدريب ٦ لعرض مجموع الأعداد التي تقبل القسمة على عدد ما يتم إدخاله في صندوق نص TextBox3 من بداية عدد ما يتم إدخاله في صندوق نص TextBox1 إلى قيمة نهاية يتم إدخالها في صندوق نص TextBox2 ووضع الناتج في مربع عنوان Label1 وذلك عند النقر على زر Button ؟

```
Public Class Form1
Private Sub Button1_Click(ByVal .....
Dim M , Start , Last , Stepp , Sum As Single
Me.TextBox4.Text = ""
Start = Me.TextBox1.Text
Last = Me.TextBox2.Text
Stepp = Me.TextBox3.Text
For M = Start To Last Step Stepp
Sum = Sum + M
Next
Me.TextBox4.Text = Sum
End Sub
End Class
```



ملحوظة هامة :

عدد مرات التكرار =  $\frac{\text{القيمة النهائية للعداد} - \text{القيمة الابتدائية للعداد}}{\text{قيمة الزيادة للعداد}} + ١$  مرة

قيمة الزيادة للعداد

قيمة M بعد إنتهاء الحلقة التكرارية = القيمة النهائية للعداد + قيمة معدل الزيادة

## ثانياً: استخدام الجملة : Do While ..... Loop

تستخدم لتكرار تنفيذ كود Code محدد لعدد من المرات غير معروف مسبقاً بناءً على شرط معين .  
لذلك هي مفيدة في حالة عدم معرفة عدد مرات التكرار مسبقاً بشكل قاطع .

الصيغة العامة لهذه الجملة

Do While (التعبير الشرطي أو الشرط) Conditional Expression

Code

الأوامر التي سيتم تكرارها  
طالما كان ناتج الشرط True

Loop

- المقصود بجملة Do While تكرار الكود Code طالما التعبير الشرطي صحيح True
- أمر التكرار Do يستخدم لتكرار مجموعة من الأوامر إذا تحقق شرط معين ( True ) ولا يعرف لها عدد مرات تكرار محدد مسبقاً وإنما ينتهي التكرار عندما لا يتحقق الشرط ( False )

**أ تدريب ٨ :** قم بتصميم وكتابة كود البرمجة والمطلوب عند النقر على زر Button1 يظهر صندوق إدخال InputBox لإدخال الإسم الأول وبعد النقر على زر OK أو ضغط مفتاح الإدخال يضاف الإسم إلى ListBox1 ويظهر صندوق إدخال InputBox مرة أخرى لإدخال الإسم الثاني وهكذا وعند إدخال كلمة "انهاء" يتم إنهاء إدخال الأسماء ؟

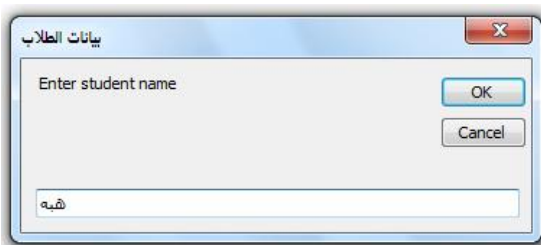
**الحل** يتم إنشاء و تصميم مشروع يحتوي على نموذج Form كما تعلمنا في الفصل الدراسي الأول ، ثم كتابة كود Code البرمجة الخاص بزر الأمر ( إدخال الأسماء ) باستخدام إجراء الحدث Button1\_Click كما يلي :

```
Public Class Form1
    Private Sub Button1_Click(ByVal ...
        Dim x As String
        Do While x <> "انهاء"
        {
            x = InputBox("Enter student name")
            Me.ListBox1.Items.Add(x)
        }
    Loop
End Sub
End Class
```



بعد النقر على زر الأمر ( إدخال الأسماء ) يظهر صندوق إدخال InputBox التالي :  
من تتبع الكود Code السابق وبعد تشغيله نجد :

هذه الحلقة التكرارية غير معلوم عدد مرات تكرارها مسبقاً ، حيث ينتهي التكرار عند إدخال قيمة معينة في صندوق إدخال InputBox وهي (انهاء)



k استخدام الدالة InputBox ووظيفتها هي استقبال قيمة من مستخدم البرنامج وإرجاع القيمة المدخلة في متغير حرفي من النوع String . حتى ولو كانت القيم التي أدخلها المستخدم قيم غير حرفية (رقمية مثلاً)  
l يحتوي المتغير x على ما تم إدخاله في صندوق إدخال InputBox .

m التعبير الشرطي المستخدم مع Do While يظل True طالما أن قيمة x لا تساوي كلمة ( انهاء ) .  
n عند إدخال كلمة ( انهاء ) يصبح الشرط False ويخرج من الحلقة التكرارية وينفذ الأوامر التي تلي Loop وهي End Sub .

○ نلاحظ إضافة كلمة ( انتهاء ) إلى صندوق القائمة ListBox1 رغم أنها شرط الخروج من الحلقة ، حيث أن أمر الإضافة قد حدث قبل الوصول إلى أول الحلقة الذي يتم فيه اختبار التعبير الشرطي ولتجنب إضافة كلمة ( انتهاء ) إلى ListBox1 نضيف شرط بعد عملية الإضافة بالأمر : If x = "إنهاء" Then Exit Do وهذا يعني عندما تصبح قيمة المتغير x تساوي كلمة ( انتهاء ) ينفذ الأمر Exit Do ( أى الخروج مباشرة من جملة Do والخروج من الحلقة التكرارية ) بدون حتى اختبار التعبير الشرطي المكتوب في سطر Do While

p الأمر Exit Do يستخدم في الخروج من جملة الحلقة التكرارية Do While من داخل الحلقة وليس من خلال سطر Do While

q عند كتابة كلمة ( انتهاء ) يجب الإلتزام بكتابتها في صندوق إدخال InputBox بنفس التهجى المكتوب في الكود ، بمعنى أن المستخدم لو كتبها بهمزة تحت الألف أو كتبها ( نهاية ) أو ( نهايه ) لن يخرج من الحلقة التكرارية ويمكن تفادى ذلك بكتابة الشرط هكذا .

r تم استخدام المعامل المنطقي And والذي يعنى تكرار ما بداخل الحلقة طالما قيمة x لا تساوي كلمة ( انتهاء ) ولا تساوي كلمة ( إنهاء ) ولا تساوي كلمة ( نهاية ) .

s لتلافى تكرار إضافة الكتابة داخل ListBox1 عند تنفيذ الإجراء مرة ثانية يجب كتابة الكود التالي :  
Me.ListBox1.Items.Clear()

وعلى ذلك يصبح الكود Code بعد تعديله كما يلى :

```
Public Class Form1
    Private Sub Button1_Click(ByVal ...
        Dim x As String
        S Me.ListBox1.Items.Clear()
        Q Do While x <> "إنهاء" And x <> "إنهاء" And x <> "نهاية" And x <> "نهاية"
            x = InputBox("Enter student name")
            O If x = "إنهاء" Or x = "إنهاء" Or x = "نهاية" Or x = "نهاية" Then Exit Do
            Me.ListBox1.Items.Add(x)
        Loop
    End Sub
End Class
```

**أ تدريب ٩ :** قم بتحويل الكود Code الخاص بالتدريب رقم ١ بحيث يتم استخدام الحلقة التكرارية Do While بدلاً من الحلقة التكرارية ؟ For

```
Public Class Form1
    Private Sub Button1_Click(ByVal ...
        Dim M As Byte
        For M = 1 To 3 Step 1
            MsgBox(M)
        Next
    End Sub
End Class
```

```
Public Class Form1
    Private Sub Button1_Click(ByVal ...
        Dim M As Byte
        M = 1
        Do While M <= 3
            MsgBox(M)
            M = M + 1
        Loop
    End Sub
End Class
```

• من عملية التحويل السابقة نلاحظ :

١ - الحلقة التكرارية For تحتوى على ثلاثة أجزاء

٢ - قيمة البداية للمتغير وهي M = 1

٣ - شرط استمرار التكرار ( نهاية التكرار ) وهو لا يتعدى القيمة 3

٤ - قيمة الزيادة في متغير الحلقة وهي Step 1

k لذا عند التحويل من جملة For إلى جملة Do While يجب كتابة الـ ٣ أجزاء السابقة كل في سطر أوامر منفصل كما ظهر في الكود

**أ تدريب ١٠ :** قم بتحويل الكود Code الخاص بالتدريب رقم ٣ بحيث يتم استخدام الحلقة التكرارية Do While بدلاً من الحلقة التكرارية For ؟

```
Public Class Form1
    Private Sub Button1_Click(ByVal ..
        Dim M, prodect As Integer
        Dim str As String

        For M = 1 To 12 Step 1
            str = 3 & "I " & M & "="
            prodect = 3 * M
            Me.TextBox1.Text = Me.TextBox1.Text _
                & str & prodect & vbCrLf

        Next
    End Sub
End Class
```

```
Public Class Form1
    Private Sub Button1_Click(ByVal ..
        Dim M, prodect As Integer
        Dim str As String

        M = 1
        Do While M <= 12
            str = 3 & "I " & M & "="
            prodect = 3 * M
            Me.TextBox1.Text = Me.TextBox1.Text _
                & str & prodect & vbCrLf
            M = M + 1
        Loop
    End Sub
End Class
```

- من عمليات التحويل السابقة نلاحظ :  
يمكن التحويل من جملة For إلى جملة Do While وبالعكس باتباع الطريقة السابقة إذا كان متغير الحلقة متغير عددي.
- يمكن التحويل من جملة For إلى جملة Do While بسهولة لأننا في جملتي التكرار استخدمنا متغير عددي كعداد (أو متغير الحلقة) ، لذا يصبح من الصعوبة بمكان التحويل من جملة Do While إلى جملة For إذا كانت جملة Do While تعتمد في شرطها على متغير غير عددي

### ثالثاً : استخدام أداة المؤقت Timer

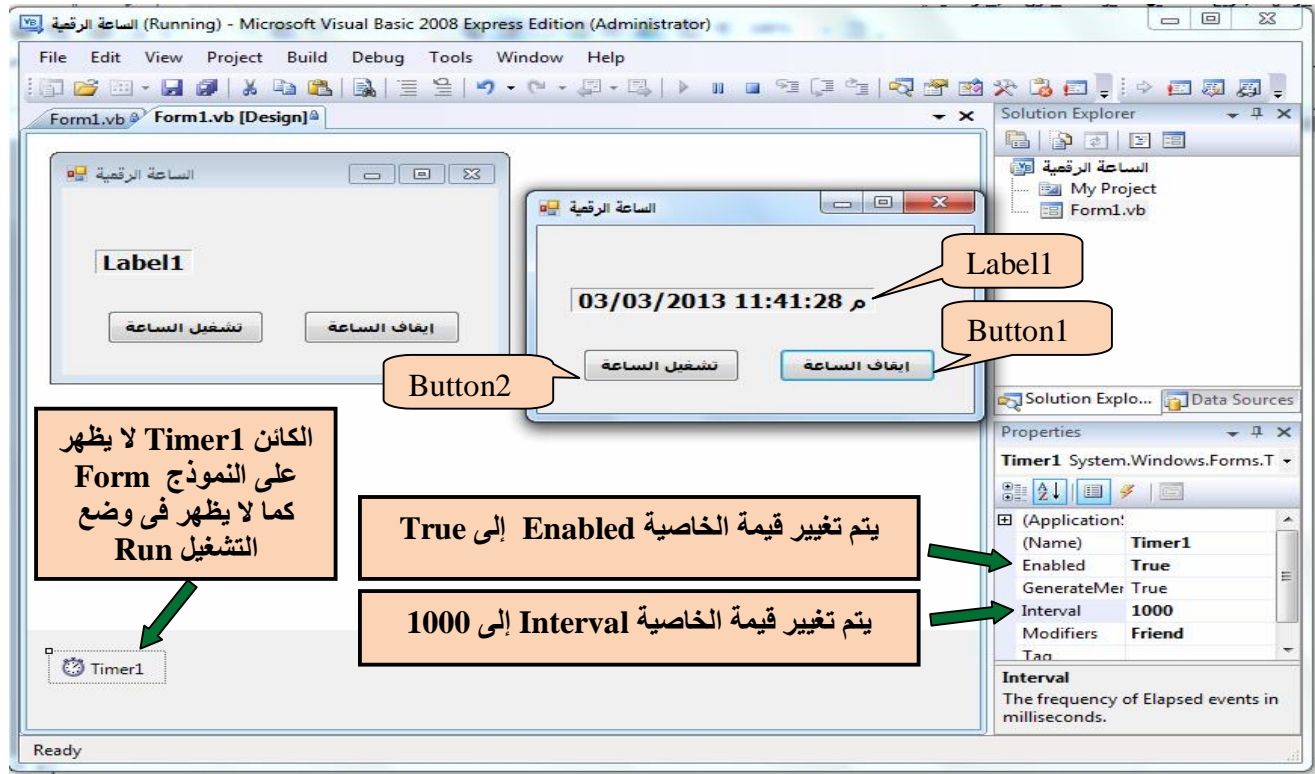
- تستخدم لتكرار تنفيذ كود Code محدد كل فترة زمنية محددة أو لتكرار تنفيذ كود Code محدد لوقت محدد.
- لذلك هي مفيدة عندما يكون تكرار تنفيذ الكود Code مرتبطاً بالوقت .
- إن تكرار الكود باستخدام المؤقت Timer يشبه الحلقات التكرارية السابقة For و Do ولكن تكرار تنفيذ هذا الكود محدد كل فترة زمنية محددة ، لذا يجب وضع شرط لإيقاف عملية التكرار أو كتابة إجراء آخر لإيقاف عملية التكرار وإلا سيكرر تنفيذ الكود إلى ما لا نهاية.
- الحدث Tick للأداة Timer هو الحدث الافتراضي لها والذي يظهر عند النقر المزدوج DblClick على الأداة Timer لكتابة الكود Code به ، وهو يعنى مرور الفترة الزمنية ( المتساوية ) المحددة في الخاصية Interval .
- عند إضافة الأداة Timer إلى النموذج Form لا بد من تحديد قيم للخصائص Properties التالية له :

الخاصية	القيمة الافتراضية	الوظيفة
Enabled	False	تمكين عمل المؤقت بجعلها True أو إيقاف عمله بجعلها False ويمكن ضبطها برمجياً من خل الكود Code : <code>Me.Timer1.Enabled = True</code>
Interval	١٠٠	تحديد الفترة الزمنية التي ينشط فيها الـ Timer بالملي ثانية ( الثانية = ١٠٠٠ ميلي ثانية )

- ينشط الحدث Tick بسبب نشاط الـ Timer على فترات زمنية متساوية هذه الفترات تسمى Interval ونتيجة لذلك ينشط الحدث Tick كما يجب أن تكون قيمة الخاصية Enabled للكائن Timer هي True
- لكي نستطيع استخدام أو تنشيط الكائن Timer كل فترات زمنية متساوية يجب تحديد قيمة موجبة للخاصية Interval وكذلك تغيير قيمة خاصية التمكين Enabled إلى True
- وحدة قياس قيمة الخاصية Interval هي الملي ثانية وكل ١٠٠٠ ميلي ثانية تساوي ثانية واحدة .
- القيمة الافتراضية للخاصية Enabled للكائن Timer هي False
- القيمة الافتراضية للخاصية Interval للكائن Timer هي 100

» تدريب ١١ : قم بتصميم وكتابة كود البرمجة لعرض ساعة رقمية مع إمكانية تشغيلها وإيقافها في أى وقت ؟

**الحل** يتم تصميم نافذة النموذج Form كما في الشكل التالي :



- يتم تغيير قيمة الخاصية Enabled للأداة Timer1 إلى True ، حتى ينشط الحدث Tick تلقائياً عند تشغيل البرنامج.
- يتم تغيير قيمة الخاصية Interval للأداة Timer1 إلى 1000 ، حتى ينشط الحدث Tick كل فترات زمنية متساوية تساوي ١ ثانية.
- يتم تغيير قيمة الخاصية BorderStyle للأداة Label1 إلى Fixed3D لكي يظهر بشكل منسق وجذاب.
- ثم يتم النقر المزدوج على الأداة Timer1 لكتابة الكود التالي في معالج الحدث Timer1\_Tick
- ثم يتم النقر المزدوج على الأداة Button1 لكتابة الكود التالي في معالج الحدث Button1\_Click
- ثم يتم النقر المزدوج على الأداة Button2 لكتابة الكود التالي في معالج الحدث Button2\_Click

```
Public Class Form1
```

```
Private Sub Timer1_Tick(ByVal .....  
    Me.Label1.Text = Now()  
End Sub
```

تخصيص ناتج الخاصية Now() للخاصية Text  
لأداة مربع العنوان Label1

```
Private Sub Button1_Click(ByVal .....  
    Me.Timer1.Enabled = False  
End Sub
```

تخصيص القيمة False للخاصية Enabled لأداة  
الموقت Timer1 لمنع وقوع الحدث Tick له

```
Private Sub Button2_Click(ByVal .....  
    Me.Timer1.Enabled = True  
End Sub
```

تخصيص القيمة True للخاصية Enabled لأداة  
الموقت Timer1 لتمكين وقوع الحدث Tick له

```
End Class
```



- الدالة Now() نأخذها هو تاريخ ووقت جهاز الكمبيوتر وهي من الدوال التي لا تأخذ وسائط بين القوسين.
- لمنع وقوع الحدث Tick برمجياً نعطي الخاصية Enabled للكائن Timer1 القيمة False ، كما في الإجراء Button1\_Click.
- عند تشغيل البرنامج ستلاحظ ظهور التاريخ والوقت بعد مرور ١٠٠٠ مللي ثانية ثم يتم تخصيص القيمة مرة أخرى بعد مرور الثانية ..... وهكذا ، حيث يتم تنفيذ كود الإجراء الخاص بالحدث Tick للأداة Timer1 كل ١ ثانية بعرض التاريخ والوقت ، لذا يبدو مربع العنوان Label1 كأنه ساعة رقمية .
- لجعل الكائن Label1 يعرض الوقت فقط بدون التاريخ يمكنك إستبدال الدالة Now() بالدالة TimeOfDay() كما يلي :

```
Private Sub Timer1_Tick(ByVal  
    Me.Label1.Text = TimeOfDay()  
End Sub
```

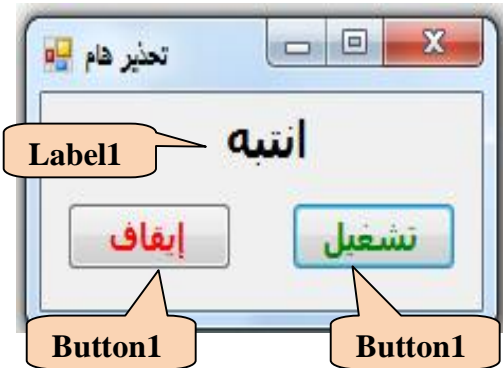


¼ تدريب ١٢ : قم بإنشاء مشروع Project يحتوي على نموذج Form التالي به أداة عنوان Label تحتوي على نص " انتبه " وعند تشغيل المشروع تظهر أداة العنوان Label لمدة نصف ثانية ثم تختفي لمدة نصف ثانية وهكذا و عند النقر على زر إيقاف تظهر أداة العنوان Label بشكل نهائي بدون أن تختفي و عند النقر على زر تشغيل تعود للظهور والإختفاء مرة أخرى ؟

**الحل** إرشادات الحل :

ز يضاف أداة عنوان Label1 ويتم تغيير خاصية Text لها إلى " انتبه " في وضع التصميم Design .  
k يضاف أداتي زر أمر Button1 و Button2 ويتم تغيير خاصية Text لهما إلى " تشغيل " و " إيقاف " في وضع التصميم Design .  
l يضاف أداة مؤقت Timer1 ويتم تغيير خاصية Enabled له إلى القيمة True في وضع التصميم Design .  
m يتم تغيير قيمة الخاصية Interval للأداة Timer1 إلى 500، حتى ينشط الحدث Tick كل فترات زمنية متساوية تساوي نصف ثانية

```
Public Class Form1  
    Private Sub Timer1_Tick(ByVal ...  
        If Me.Label1.Visible = True Then  
            Me.Label1.Visible = False  
        Else  
            Me.Label1.Visible = True  
        End If  
    End Sub  
    Private Sub Button1_Click(ByVal  
        Me.Timer1.Enabled = True  
    End Sub  
    Private Sub Button2_Click(ByVal  
        Me.Timer1.Enabled = False  
    End Sub  
End Class
```



n يمكن تغيير كود الإجراء Timer1\_Tick ليصبح أكثر بساطة بكتابة هذا الكود فقط :  
Me.Label1.Visible = Not(Me.Label1.Visible)  
— حيث تم استخدام المعامل المنطقي Not والذي يستخدم للحصول على عكس ناتج ما بداخل القوسين ، مع ملاحظة أن ما بداخل القوسين تعبير شرطي نأخذه ( True أو False )  
— فإذا كان ما بداخل القوسين Not (True) كان الناتج False وإذا كان ما بداخل القوسين Not (False) كان الناتج True



١/٢ تدريب ١٣ : قم بإنشاء مشروع Project يحتوي على نموذج Form وعند إدخال رقم في صندوق نص TextBox1 ثم انقر على زر (الأعداد الفردية) يتم عرض الأعداد الفردية مرتبة تصاعدياً داخل صندوق القائمة ListBox1 حتى الرقم المدخل ، أما عند النقر على زر (الأعداد الزوجية) يتم عرض الأعداد الزوجية مرتبة تصاعدياً داخل صندوق القائمة ListBox1 حتى الرقم المدخل ؟

**الحل**



Do While باستخدام جملة

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        i = 1
        Do While i <= N
            Me.ListBox1.Items.Add(i)
            i = i + 2
        Loop
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        i = 2
        Do While i <= N
            Me.ListBox1.Items.Add(i)
            i = i + 2
        Loop
    End Sub
End Class
```

كود الأعداد  
الفردية

كود الأعداد  
الزوجية



For... Next باستخدام جملة

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()

        For i = 1 To N Step 2
            Me.ListBox1.Items.Add(i)

        Next
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()

        For i = 2 To N Step 2
            Me.ListBox1.Items.Add(i)

        Next
    End Sub
End Class
```

نلاحظ في التدريب السابق أن ١ - متغير الحلقة التكرارية في كود الأعداد الفردية بدأ بعدد فردي هو

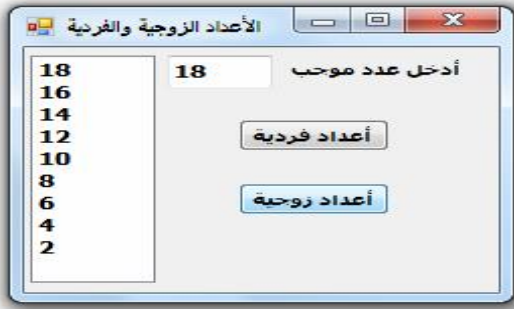
٢ - متغير الحلقة التكرارية في كود الأعداد الزوجية بدأ بعدد زوجي هو

- من الممكن أن يتم الإعلان عن المتغيرات Dim N, i As Integer مرة واحدة على مستوى أو نطاق التصنيف بدلاً من تكرارهما في كل إجراء

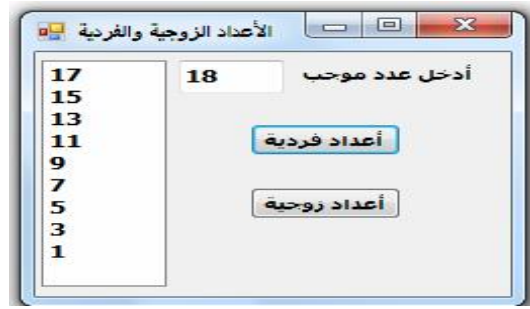
- لتلافي تكرار إضافة الكتابة داخل ListBox1 عند تنفيذ أي من الإجراءات ثانياً يجب كتابة الكود التالي  
Me.ListBox1.Items.Clear() والذي يستخدم لمسح محتويات صندوق القائمة ListBox1.

٣/٤ تدريب ١٤ : قم بتعديل كود البرنامج في التدريب ١٣ السابق بحيث يتم طباعة الأعداد تنازلياً بدلاً من تصاعدياً ؟

الحل



باستخدام جملة Do While



باستخدام جملة For... Next

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        If N Mod 2 = 0 Then
            N = N - 1
        End If
        i = N
        Do While i >= 1
            Me.ListBox1.Items.Add(i)
            i = i - 2
        Loop
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        If N Mod 2 <> 0 Then
            N = N - 1
        End If
        i = N
        Do While i >= 2
            Me.ListBox1.Items.Add(i)
            i = i - 2
        Loop
    End Sub
End Class
```

كود الأعداد  
الفردية

كود الأعداد  
الزوجية

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        If N Mod 2 = 0 Then
            N = N - 1
        End If
        For i = N To 1 Step -2
            Me.ListBox1.Items.Add(i)
        Next
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i As Integer
        N = Me.TextBox1.Text
        Me.ListBox1.Items.Clear()
        If N Mod 2 <> 0 Then
            N = N - 1
        End If
        For i = N To 2 Step -2
            Me.ListBox1.Items.Add(i)
        Next
    End Sub
End Class
```

نلاحظ في التدريب السابق أن :

ز تم إضافة جملة If إضافية في كود الأعداد الفردية للتحقق من أن العدد N المدخل في صندوق النص TextBox1 عدد زوجي أم لا أي يقبل القسمة على 2 فإذا كان ناتج الشرط True يتم إنقاص قيمة N ب 1 حتى نحصل على عدد فردي.  
k تم إضافة جملة If إضافية في كود الأعداد الزوجية للتحقق من أن العدد N المدخل في صندوق النص TextBox1 عدد فردي أم لا أي لا يقبل القسمة على 2 فإذا كان ناتج الشرط True يتم إنقاص قيمة N ب 1 حتى نحصل على عدد زوجي.  
1 نلاحظ أن متغير الحلقة التكرارية i في جميع الإجراءات السابقة بدأ بقيمة ابتدائية هي N وانتهى بقيمة نهاية هي العدد 1 في إجراء العدد الفردي و العدد 2 في إجراء العدد الزوجي ، ومعدل الزيادة لهذا المتغير i أخذ قيمة سالبة هي -2

ن تدريب ١٥ : قم بإنشاء مشروع Project يحتوي على نموذج Form وعند إدخال رقم في صندوق نص TextBox1 ثم انقر على زر (مجموع الأعداد الفردية) يتم عرض مجموع الأعداد الفردية من ١ حتى الرقم المدخل داخل مربع عنوان Label3، أما انقر على زر (مجموع الأعداد الزوجية) يتم عرض مجموع الأعداد الزوجية من ٢ حتى الرقم المدخل داخل مربع عنوان Label3؟

باستخدام جملة Do While

باستخدام جملة For... Next

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i, Sum As Integer
        N = Me.TextBox1.Text
        i = 1
        Do While i <= N
            Sum = Sum + i
            i = i + 2
        Loop
        Me.Label3.Text = Sum
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i, Sum As Integer
        N = Me.TextBox1.Text
        i = 2
        Do While i <= N
            Sum = Sum + i
            i = i + 2
        Loop
        Me.Label3.Text = Sum
    End Sub
End Class
```

كود مجموع  
الأعداد الفردية

كود مجموع  
الأعداد الزوجية

```
Public Class Form1
    Private Sub Button1_Click(ByVal
        Dim N, i, Sum As Integer
        N = Me.TextBox1.Text
        For i = 1 To N Step 2
            Sum = Sum + i
        Next
        Me.Label3.Text = Sum
    End Sub
    Private Sub Button2_Click(ByVal
        Dim N, i, Sum As Integer
        N = Me.TextBox1.Text
        For i = 2 To N Step 2
            Sum = Sum + i
        Next
        Me.Label3.Text = Sum
    End Sub
End Class
```

نلاحظ في التدريب السابق أنه تم الإعلان عن متغير جديد هو Sum وهو المخزن الذي سيضاف إليه في كل مرة يتم فيها

التكرار قيمة المتغير i بجملة التخصيص Sum = Sum + i

k تم إضافة سطر الأوامر Me.Label3.Text = Sum خارج الحلقة التكرارية ( أي بعد الإنتهاء من تنفيذ الحلقة

التكرارية ومن خلاله يتم تخصيص قيمة المخزن Sum للخاصية Text للأداة Label3

l من الممكن أن يتم الإعلان عن المتغيرات Dim N, I, Sum As Integer مرة واحدة على مستوى أو نطاق التصنيف

بدلاً من تكرارهما في كل إجراء

## الفصل الرابع ( الإجراءات Procedures )

**مقدمة :** عند اضافة نافذة نموذج Form جديدة للمشروع ينشأ تصنيف جديد اسمه مطابق لإسم النموذج Form1 : وفي نطاق هذا التصنيف يتم :

✓ الإعلان عن إجراءات الأحداث Event procedures أو معالجات الأحداث مثل اجراء الحدث الخاص بزر معين مثلاً .

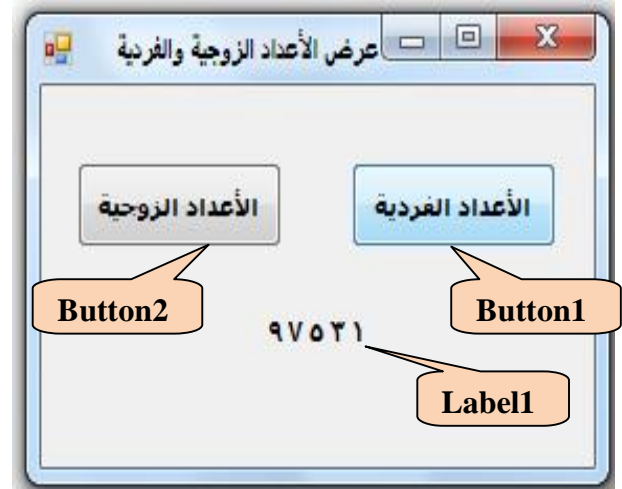
✓ كذلك الإعلان عن المتغيرات variables والثوابت Constant وكذلك الإعلان عن مايسمى بـ Block مثل جملة If...then و جملة For... next وغيرها .

✓ من الممكن أن يكون هذا الإعلان على مستوى التصنيف Class ( نطاق التصنيف ) أو في نطاق الإجراء ( مثل الإعلان عن متغير معين داخل اجراء حدث زر معين ) . كما بالمثال التالي :

• **تدريب ١ :** قم بتصميم وكتابة كود البرمجة اللازم لعرض الأعداد الفردية و الأعداد الزوجية من ١ إلى ١٠ داخل مربع عنوان Label وذلك عند الضغط على زرى أمر Buttons ؟

**الحل**

```
Public Class Form1
    Dim total As Integer
    Private Sub Button1_Click(ByVal ...
        'عرض الأعداد الفردية من ١ إلى ١٠
        Dim i As Integer
        Me.Label1.Text = ""
        For i = 1 To 10 Step 2
            Me.Label1.Text =
                Me.Label1.Text & " " & i
        Next
    End Sub
    Private Sub Button2_Click(ByVal ...
        'عرض الأعداد الزوجية من ١ إلى ١٠
        Dim i As Integer
        Me.Label1.Text = ""
        For i = 2 To 10 Step 2
            Me.Label1.Text =
                Me.Label1.Text & " " & i
        Next
    End Sub
End Class
```



من الكود السابق نجد أن :

١ الإعلان عن تصنيف Class تحت اسم Form1 .

٢ الإعلان عن متغير Variable تحت اسم total على مستوى ( نطاق ) التصنيف Form1 . ليس له فائدة في هذا البرنامج

٣ الإعلان عن إجراءات للأحداث تحت مستوى ( نطاق ) التصنيف Form1 ، تحت اسم Button1\_Click و Button2\_Click .

٤ الإعلان عن متغيرين Variables بنفس الإسم داخل الإجراءات Button1\_Click و Button2\_Click ، تحت اسم i .

٥ الإعلان عن Block أو بناء جملة For داخل الإجراءات Button1\_Click و Button2\_Click .

من الملاحظ تكرار محتوى إجراءات الأحداث Button1\_Click و Button2\_Click من أوامر ما عدا القيمة الابتدائية للعداد i ، وفي السطور التالية سيتم استخدام طريقة جديدة لكتابة إجراء معين واستدعائه في إجراءات أخرى أكثر من مرة لتلافي مشكلة التكرار ولتسهيل فهم الأوامر " وهى ما تسمى الإجراءات "

**الإجراء Procedure :** هو مجموعة من الأوامر والتعليمات Code تحت اسم ما ، عند استدعاء هذا الاسم يتم تنفيذ

هذه الأوامر والتعليمات . **وله نوعان** إما أن تكون Sub لا تعود بقيمة k أو Function تعود بقيمة الإجراء Procedure بصفة عامة يعلن عنه مرة واحدة ( تحت أحد الصورتين السابقتين ) ويمكن استدعاؤها العديد من المرات وهذا يوفر تكرار كتابة الكود Code فى أماكن استدعاء هذه الإجراءات Procedures .

### الإعلان عن الإجراء Sub :

نلجأ للإعلان عن الإجراء Sub فى حالة إذا ما كان لدينا كود Code سيتكرر كتابته فى أكثر من موضع داخل التصنيف Class وكذلك بهدف تنظيم كتابة الكود Code حتى يسهل قراءته وفهمه ومن ثم تعديله إذا لزم الأمر .

Sub Name ( Parameters )  
Code  
End Sub

الصيغة العامة للإعلان عن الإجراء Sub

ز يبدأ الإعلان عن الإجراء Sub بكلمة Sub .

Name k تعبر عن اسم الإجراء Sub .

Parameters l ( المعاملات أو الوسائط ) وهى عبارة عن القيم التى سوف تستقبل عند استدعاء الإجراء Procedure

والتي تستخدم فى الكود Code وله نوع بيان Data Type .

Code m مجموعة من الأوامر والتعليمات التى ستنفذ عند استدعاء الإجراء Sub

• تم وضع ملاحظة فى التدريب رقم 1 السابق والتي تفيد بأن الكود Code المكتوب داخل كل من إجراء التحديث Button1\_Click و Button2\_Click متكرر ما عدا القيمة الابتدائية لعداد التكرار ( i ) حيث فى كود الأعداد الزوجية ظهر بالقيمة 2 وفى كود الأعداد الفردية ظهر بالقيمة 1 .

**تدريب ٢ :** قم بتعديل كود البرمجة فى تدريب 1 السابق باستخدام الإعلان عن إجراء Procedure من نوع Sub يستخدم فى إجراء التحديث المستخدمين لعرض الأعداد الفردية والأعداد الزوجية من 1 إلى 10 داخل مربع عنوان Label وذلك عند الضغط على زرى أمر Buttons ؟

الإعلان عن معامل Parameter

```
Public Class Form1
    j Sub ShowOddOrEven(ByVal Start As Integer)
        Dim i As Integer
        Me.Label1.Text = ""
        k For i = Start To 10 Step 2
            Me.Label1.Text = Me.Label1.Text & " " & i
        Next
    End Sub
    Private Sub Button1_Click(ByVal ...
        عرض الأعداد الفردية من 1 إلى 10
        l ShowOddOrEven(1) m
    End Sub
    Private Sub Button2_Click(ByVal ...
        عرض الأعداد الزوجية من 1 إلى 10
        l ShowOddOrEven(2) m
    End Sub
End Class
```

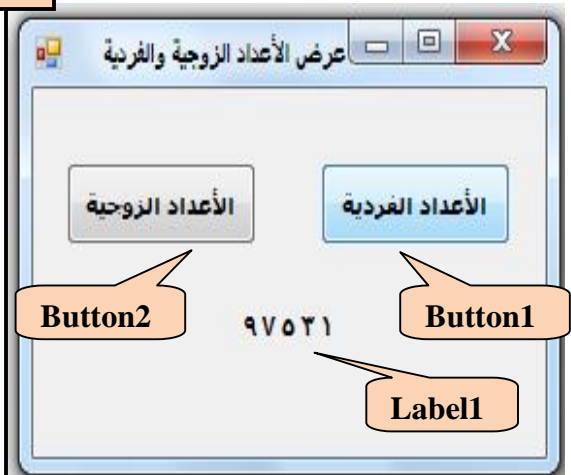
Sub الإعلان عن الإجراء

الكود الذى ينفذ عند استدعاء الإجراء Sub

Sub الإعلان عن الإجراء

Sub الإعلان عن الإجراء

Argument تحديد قيمة



من الكود السابق نجد أن :

1 الإعلان عن إجراء Sub تحت اسم ShowOddOrEven له معامل أو وسيط واحد تحت اسم Start حتى يمكن تحديد قيمة بداية عدد تكرار الحلقة التكرارية For وبناءً عليه تعرض الأعداد الفردية التي تبدأ بالعدد 1 أو الأعداد الزوجية التي تبدأ بالعدد 2.

k الإعلان عن معامل أو وسيط Prameter واحد تحت اسم Start من نوع عددي Integer في الإجراء ShowOddOrEven().

l استدعاء الإجراء Sub المسمى ShowOddOrEven داخل إجراء الحدث Button1\_Click مع استخدام معامل Prameter بقيمة Argument هي 1 حتى تكون قيمة بداية عدد الحلقة هو 1 بهذا الأمر ShowOddOrEven(1) لطباعة الأعداد الفردية التي تبدأ بالعدد 1.

كما تم استدعاء نفس الإجراء Sub داخل إجراء الحدث Button2\_Click مع استخدام معامل Prameter بقيمة Argument هي 2 حتى تكون قيمة بداية عدد الحلقة هو 2 بهذا الأمر ShowOddOrEven(2) لطباعة الأعداد الزوجية التي تبدأ بالعدد 2.

m يجب أن يكون قيم الوسيط Argument 1 و 2 من نوع بيان متوافق مع نوع بيان الوسيط Prameter المسمى Start.

لأن عدم استخدام الإعلان عن المعامل أو الوسيط Prameter المسمى Start عند الإعلان عن إجراء Sub تحت اسم

ShowOddOrEven() بالشكل التالي :

```
Sub ShowOddOrEven ( )
    Dim i As Integer
    Me.Label1.Text = ""
    For i = 1 To 10 Step 2
        Me.Label1.Text = Me.Label1.Text & " "
    & i
    Next
End Sub
```

يحدث الآتي :

عند اختبار البرنامج نجد أن النقر على زر الأعداد الفردية و زر الأعداد الزوجية سوف يعطي نفس النتيجة وهي طباعة الأعداد الفردية لكلا الزرين ، والسبب أن قيمة بداية عدد الحلقة التكرارية For واحدة وهي القيمة 1 ،

لذا تم الإعلان عن المعامل أو الوسيط Prameter المسمى Start واستخدامه بقيمة Argument ShowOddOrEven(1) و

ShowOddOrEven(2) كما شرحنا

لا يمكن الإعلان عن إجراء Procedure بدون استخدام أي معامل أو وسيط Prameter أو باستخدام معامل أو أكثر لا تستخدم المعاملات أو الوسائط Parameters كوسيلة لإستقبال قيم Argument من خارج الإجراء هذه القيم غير معلومة مسبقاً إنما تحدد عند استدعاء هذا الإجراء Procedure كما حدث في إجراءات الحدث للأعداد الفردية والزوجية لا هذه القيم Argument يمكن أن تكون قيم مجردة أو قيمة من متغير أو ثابت أو دالة ( سيأتى ذكرها لاحقاً ) .



1 تدريب ٣ :

قم بتطوير كود البرمجة في تدريب ٢ السابق بحيث يكون الإجراء ShowOddOrEven() يستقبل قيمة بداية Start وقيمة نهاية Last في عملية التكرار؟

الإعلان عن الإجراء Sub

الإعلان عن معاملين Parameters

```

Public Class Form1
    Sub ShowOddOrEven(ByVal Start As Integer , ByVal Last As Integer)
        Dim i As Integer
        Me.Label1.Text = ""
        For i = Start To Last Step 2
            Me.Label1.Text = Me.Label1.Text & " " & i
        Next
    End Sub
    Private Sub Button1_Click(ByVal ...
        'عرض الأعداد الفردية من ١ إلى ١٠
        ShowOddOrEven(1, 10)
    End Sub
    Private Sub Button2_Click(ByVal ...
        'عرض الأعداد الزوجية من ١ إلى ١٠
        ShowOddOrEven(2, 10)
    End Sub
End Class
    
```

استخدام المعاملين Parameters

استدعاء الإجراء وتحديد قيم Argument لها

لا يتم الإعلان عن الإجراء Procedure مرة واحدة فقط ثم يمكن إستدعائه أى عدد من المرات .  
 لاحظت أننا يمكننا التحكم في عرض الأعداد سواء كانت فردية أو زوجية ، وكذلك نطاق أو مدى الأعداد ، فمثلاً لو أعطينا قيم Argument هي ( 5 , 35 ) ستعرض الأعداد الفردية من ٥ : ٣٥ ، وذلك باستخدام الأمر ShowOddOrEven(5, 35).

٥ تدريب ٤ :

قم بتطوير كود البرمجة في تدريب ٣ السابق بحيث يكون الإجراء ShowOddOrEven() يستقبل قيمة بداية Start وقيمة نهاية Last وقيمة زيادة Increment في عملية التكرار؟

الإعلان عن ثلاث معاملات Parameters

```

Public Class Form1
    Sub ShowOddOrEven(ByVal Start As Integer , ByVal Last As Integer , ByVal Increment As Integer)
        Dim i As Integer
        Me.Label1.Text = ""
        For i = Start To Last Step Increment
            Me.Label1.Text = Me.Label1.Text & " " & i
        Next
    End Sub
    Private Sub Button1_Click(ByVal ...
        'عرض الأعداد الفردية من ١ إلى ١٠
        ShowOddOrEven(1, 10, 2)
    End Sub
    Private Sub Button2_Click(ByVal ...
        'عرض الأعداد الزوجية من ١ إلى ١٠
        ShowOddOrEven(2, 10, 2)
    End Sub
End Class
    
```

استخدام المعاملات Parameters

تحديد قيم Argument

لاحظت أننا يمكننا التحكم في عرض الأعداد على سبيل المثال الأعداد التي تقبل القسمة على ٧ من ١٤ إلى ٥٠٠ .  
 وذلك باستدعاء الإجراء وتخصيص قيمة Argument له كما يلي : ShowOddOrEven(14, 500, 7)



» تدريب ٥ : قم بتطوير كود البرمجة في تدريب ٤ السابق بحيث يكون الإجراء ShowNumbers() يستقبل قيمة

بداية Start وقيمة نهاية Last وقيمة زيادة Increment من خلال ثلاث صناديق نص TextBox في عملية التكرار وعرض الناتج في داخل صندوق قائمة ListBox ؟

```
Public Class Form1
    Sub ShowNumbers(ByVal Start As Integer , ByVal Last As Integer , ByVal Increment As Integer)
        Dim i As Integer
        Me.ListBox1.Items.Clear()

        For i = Start To Last Step Increment
            Me.ListBox1.Items.Add(i)
        Next
    End Sub

    Private Sub Button1_Click(ByVal ...
        ' عرض الأعداد
        Dim No1 As Integer = Me.TextBox1.Text
        Dim No2 As Integer = Me.TextBox2.Text
        Dim No3 As Integer = Me.TextBox3.Text
        ShowNumbers(No1, No2, No3)
    End Sub
End Class
```



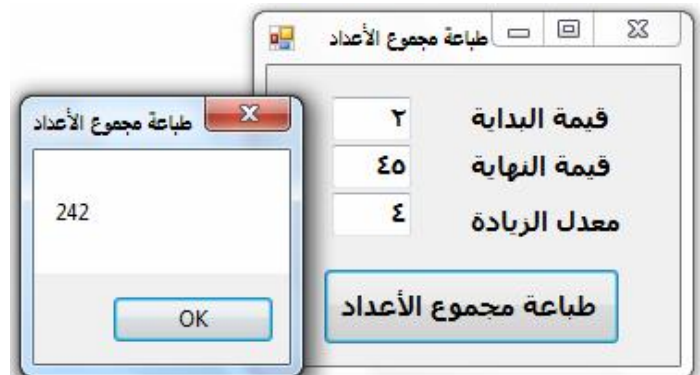
¼ تدريب ٦ : قم بتطوير كود البرمجة في تدريب ٥ السابق بحيث يكون الإجراء ShowSumNo() يستقبل قيمة بداية

Start وقيمة نهاية Last وقيمة زيادة Increment من خلال ثلاث صناديق نص TextBox في عملية التكرار وعرض ناتج مجموع الأعداد من البداية إلى النهاية في صندوق رسالة MsgBox ؟

```
Public Class Form1
    Sub ShowSumNo(ByVal Start As Integer , ByVal Last As Integer , ByVal Increment As Integer)
        Dim i, Sum As Integer

        For i = Start To Last Step Increment
            Sum = Sum + i
        Next
        MsgBox(Sum)
    End Sub

    Private Sub Button1_Click(ByVal ...
        ' عرض مجموع الأعداد
        Dim No1 As Integer = Me.TextBox1.Text
        Dim No2 As Integer = Me.TextBox2.Text
        Dim No3 As Integer = Me.TextBox3.Text
        ShowSumNo(No1, No2, No3)
    End Sub
End Class
```



## الإعلان عن الدالة Function :

نلجأ للإعلان عن دالة Function في حالة إذا ما كان لدينا Code سينتج ( تعود ) منه قيمة نحتاجها . مثال على ذلك قيمة محيط دائرة أو مساحة مربع أو صافي أجر العامل أو الضريبة المستحقة ..... إلخ ، حتى يمكن استخدامها أثناء تنفيذ تعليمات البرنامج أو إخراجها للمستخدم .

**Function** Name ( Parameters ) **As** Data Type  
**Code**  
**Return** Value  
**End Function**

الصيغة العامة للإعلان عن الدالة Function

j Function بداية الإعلان ( التعريف ) عن الدالة .

k Name تعبر عن اسم الدالة Function المقترح ويفضل أن يكون الاسم متعلق بوظيفة الدالة أو محتواها ولها نوع بيان .

l Parameters تمثل الوسائط التي سوف تستخدم في الكود وله نوع بيان Data Type كما درسنا في الإجراء Sub .

m Data Type تحدد نوع بيان الدالة الذي يوافق نوع البيان الخاصة بالقيمة الراجعة ( Value ) من الدالة Function .

n Code مجموعة الأوامر والتعليمات التي ستنفذ عند استدعاء الدالة Function .

o Value القيمة الراجعة ( العائدة ) من الدالة Function والتي يجب الإعلان عنها داخل الدالة بنوع بيان يوافق نوع البيان

الخاصة بالدالة Function ( Name ) .

p End Function نهاية الإعلان ( التعريف ) عن الدالة .

1/2 تدريب ٧ : قم بتصميم وكتابة كود البرمجة اللازم لحساب مجموع رقمين يتم ادخالهم من خلال صندوق نص TextBox واظهار الناتج في مربع عنوان Label وذلك عند النقر على زر أمر Button وذلك من خلال الإعلان عن دالة لحساب هذا المجموع ؟

```
Public Class Form1
    Function Sum(ByVal First As Single, ByVal Second As Single) As Single
        Dim Total As Single
        Total = First + Second
        Return Total
    End Function

    Private Sub Button1_Click(ByVal ...
        Dim X As Single = Me.TextBox1.Text
        Dim Y As Single = Me.TextBox2.Text
        Me.Label4.Text = Sum(X, Y)
    End Sub
End Class
```

الإعلان عن معاملي Parameters

استخدام المعاملي Parameters

القيمة الراجعة

استدعاء الدالة وتحديد قيم لها Argument

Q من الكود السابق نجد أن :

z الإعلان عن الدالة Sum من النوع العددي الغير صحيح Single وتستقبل قيمتين هما First و Second .

k الإعلان عن معاملي ( وسيطين ) للدالة Sum هما First و Second من النوع العددي الغير صحيح Single .

l الإعلان عن المتغير Total من النوع العددي الغير صحيح Single الذي يوافق نوع بيان الدالة Sum ، وهو يمثل القيمة الراجعة ( العائدة ) من الدالة Sum عند استدعائها .

m تم تخصيص ناتج جمع القيمتين First و Second للمتغير Total بالأمر : Total = First + Second .

n إرجاع قيمة المتغير Total باستخدام الأمر Return .

o الإعلان عن المتغيرين X و Y من النوع العددي Single داخل إجراء الحدث Button1\_Click ، وتخصيص مدخلات المستخدم لكل منهما من صندوق النص TextBox1 و TextBox2 على الترتيب .

p تم تخصيص القيمة الراجعة من الدالة Sum للخاصية Text للأداة Label4 وذلك بعد استقبال القيمتين X و Y كـ Argument للدالة Sum .

© من الضروري أن يكون المتغيرين من نوع بيان يوافق نوع بيان وسائط أو معاملات Parameters الدالة Function .

© ليس من الضروري أن يكون نوع بيان Data Type للمعاملات ( الوسائط ) من نفس نوع بيان Data Type الدالة أو القيمة الراجعة .

© يفضل تسمية الإجراء Sub و الدالة Function تسمية متعلقة بوظيفتها أو محتواها .

### مقارنة بين المتغيرات Variables والثوابت Constants والإجراء Procedure والدالة Function :

المتغيرات Variables	الثوابت Constants	الإجراء Sub	الدالة Function
يمكن تخصيص قيمة لها أثناء الإعلان عنها و أثناء سير (تشغيل) البرنامج	يجب تخصيص قيمة لها أثناء الإعلان عنها	لا يمكن تخصيص قيمة لها ، وإنما تستدعى باسمها فتنفذ الأوامر التي بها ،	لا يمكن تخصيص قيمة لها ، وإنما تستدعى باسمها فتنتج قيمة تخزن بها
يمكن استخدام القيم المخزنة بها في الذاكرة	يمكن استخدام القيم المخزنة بها في الذاكرة	ليس لها قيمة مخزنة في الذاكرة لأنها لا تعود بقيمة وليس لها ناتج	يمكن استخدام القيم المخزنة بها في الذاكرة و لها ناتج
لها نوع بيان Data Type مناسب	لها نوع بيان Data Type مناسب	ليس لها نوع بيان Data Type لأنها لا تعود بقيمة	لها نوع بيان Data Type هو نفس نوع بيان القيمة الراجعة
تكتب في جمل التخصيص سواء على يمين ويسار معامل التخصيص	تكتب في جمل التخصيص على يمين معامل التخصيص فقط	لا يجوز أن تكتب في أى جمل تخصيص نهائياً	تكتب في جمل التخصيص على يمين معامل التخصيص فقط ، لأنها لا يخصص لها قيمة
ليس لها وسائط أو معاملات	ليس لها وسائط أو معاملات	قد يكون لها وسائط ومعاملات قد تكون قيم مجردة أو متغيرات أو ثوابت أو دوال	قد يكون لها وسائط ومعاملات قد تكون قيم مجردة أو متغيرات أو ثوابت
لا يجوز أن تكتب كما هي في سطر أوامر بحد ذاتها	لا يجوز أن تكتب كما هي في سطر أوامر بحد ذاتها	تكتب في سطر أوامر بحد ذاتها	لا يجوز أن تكتب كما هي في سطر أوامر بحد ذاتها

3/4 تدريب ٨ : قم بتصميم وكتابة كود البرمجة اللازم لحساب مضروب العدد ٨ واظهار الناتج في صندوق رسالة MsgBox وذلك عند النقر على زر أمر Button وذلك من خلال الإعلان عن دالة لحساب هذا المضروب ؟

```
Public Class Form1
    Function Factorial(ByVal Number As Integer) As Integer
        Dim i As Integer
        Dim res As Integer = 1
        For i = 1 To Number
            res = res * i
        Next
        Return res
    End Function
    Private Sub Button1_Click(ByVal ...
        MsgBox.Show(Factorial(8))
    End Sub
End Class
```



من الكود السابق نجد أن :

- ١ الإعلان عن الدالة Factorial من النوع العددي Integer وتستقبل قيمة ( معامل أو وسيط ) واحدة هي Number.
- ٢ k الإعلان عن معامل ( وسيط ) واحد للدالة Factorial هو Number من النوع العددي Integer.
- ٣ l الإعلان عن المتغير i كعداد لعملية التكرار من النوع العددي Integer.
- ٤ m الإعلان عن المتغير res لتخزين ناتج حساب مضروب العدد من النوع العددي Integer الذي يوافق نوع بيان الدالة Sum ، وهو يمثل القيمة الراجعة ( العائدة ) من الدالة Factorial عند استدعائها.
- ٥ n تم إظهار القيمة الراجعة ( العائدة ) من الدالة Factorial في صندوق رسالة MsgBox بعد استدعاء الدالة ، وذلك بعد استقبال القيمة ٨ ك Argument للدالة Factorial.
- ٦ o تم استخدام الدالة المعرفة Show() المعلنة داخل التصنيف MsgBox والتي من خلالها يمكن إظهار صندوق رسالة MsgBox يتحدد محتوياته حسب الوسائط المعطاه للدالة Show() .

٧ في تدريب ٩ : قم بتطوير الكود السابق في التدريب ٨ لحساب مضروب عدد ما يتم إدخاله في صندوق نص TextBox واظهار الناتج في صندوق رسالة MsgBox وذلك عند النقر على زر أمر Button وذلك من خلال الإعلان عن دالة لحساب هذا المضروب ؟

```
Public Class Form1
    Function Factorial(ByVal Number As Integer) As Integer
        Dim i As Integer
        Dim res As Integer = 1
        For i = 1 To Number
            res = res * i
        Next
        Return res
    End Function
    Private Sub Button1_Click(ByVal ...
        Dim No As Integer = Me.TextBox1.Text
        MsgBox.Show(Factorial(No))
    End Sub
End Class
```



**أ تدريب ١٠ :** قم بتصميم وكتابة كود البرمجة اللازم لحساب مساحة دائرة نصف قطرها ٥ سم واطهار الناتج في صندوق رسالة MsgBox وذلك عند النقر على زر أمر Button وذلك من خلال الإعلان عن دالة لحساب المساحة ؟

```
Public Class Form1
    Function Area(ByVal radius As Single) As Single
        Const pi As Single = 22 / 7
        Dim res As Single
        res = pi * radius ^ 2
        Return res
    End Function

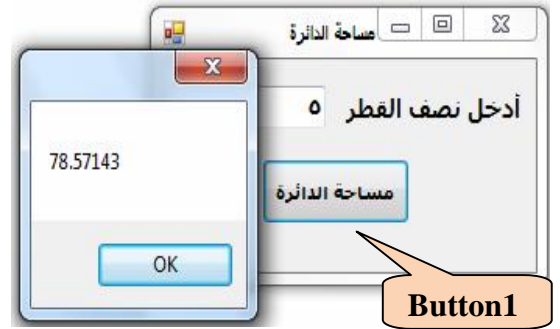
    Private Sub Button1_Click(ByVal ...
        MessageBox.Show(Area(5))
    End Sub
End Class
```



**أ تدريب ١١ :** قم بتطوير الكود السابق في التدريب ١٠ لحساب مساحة دائرة نصف قطرها عدد ما يتم إدخاله في صندوق نص TextBox واطهار الناتج في صندوق رسالة MsgBox وذلك عند النقر على زر أمر Button وذلك من خلال الإعلان عن دالة لحساب المساحة ؟

```
Public Class Form1
    Function Area(ByVal radius As Single) As Single
        Const pi As Single = 22 / 7
        Dim res As Single
        res = pi * radius ^ 2
        Return res
    End Function

    Private Sub Button1_Click(ByVal ...
        Dim Rad As Single = Me.TextBox1.Text
        MessageBox.Show(Area(radius))
    End Sub
End Class
```



### الدوال المعرفة : Predefined Function

هي دوال معرفة لدى لغة البرمجة يتم استدعاؤها مباشرة دون الحاجة للإعلان عنها ، ويتم استرجاع قيمة منها.

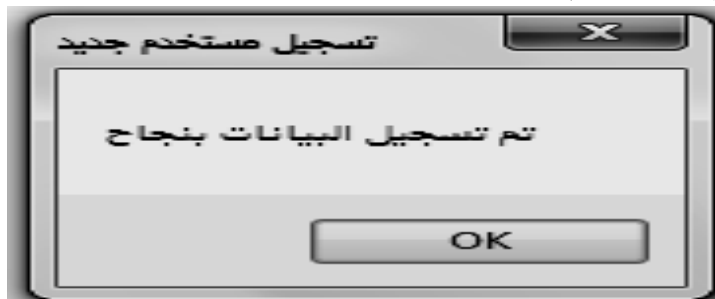
### أولاً الدالة Show() المعلنة داخل التصنيف MessageBox :

من خلال هذه الدالة يمكن إظهار صندوق رسالة MessageBox يتحدد محتوياته حسب الوسائط المعطاه للدالة Show().

مثال : (" تسجيل مستخدم جديد" , "تم تسجيل البيانات بنجاح" ) MessageBox.Show()

حيث "تم تسجيل البيانات بنجاح" يمثل محتوى صندوق الرسالة وهو وسيط إجباري.

حيث " تسجيل مستخدم جديد" يمثل عنوان صندوق الرسالة ، وهو اختياري ( يمكن عدم كتابته ).



**ثانياً الدالة ( ) IsNumeric :**

يمكن من خلالها اختبار قيمة هل هي رقمية أم لا وناتجها إما True في حالة إذا كانت القيمة بين القوسين رقمية أو False في حالة إذا كانت القيمة بين القوسين غير رقمية.

© استدعاؤها : `Me.Label1.Text = IsNumeric("five")`

عند استدعاؤها يظهر الناتج False داخل مربع العنوان Label1 لأن القيمة "five" ليست رقمية ولا يمكن تحويلها لقيمة رقمية من خلال التحويل الضمني.

© استدعاؤها : `Me.Label1.Text = IsNumeric(5)`

عند استدعاؤها يظهر الناتج True داخل مربع العنوان Label1 لأن القيمة 5 رقمية .

© استدعاؤها : `Me.Label1.Text = IsNumeric("5")`

عند استدعاؤها يظهر الناتج True داخل مربع العنوان Label1 على الرغم من أن القيمة "5" ليست رقمية ولكن أمكن تحويل تلك القيمة الحرفية إلى قيمة رقمية من خلال التحويل الضمني.

**ثالثاً الدالة ( ) Now :**

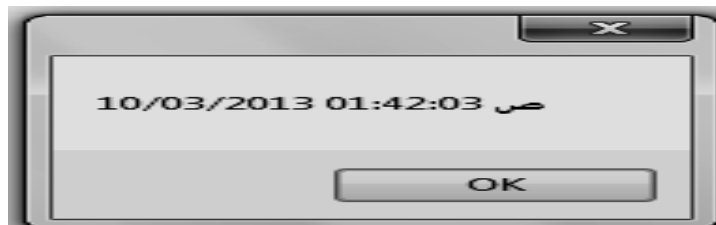
يمكن من خلالها استعراض التاريخ والوقت المسجل بنظام الكمبيوتر .

© استدعاؤها :

`Me.Label1.Text = Now()`

`MessageBox.Show(Now())`

`MsgBox(Now())`



ن جميع الدوال Function تستخدم في الطرف الأيمن فقط من معادلة التخصيص للحصول على ناتجها.

ن هناك دوال لا تأخذ أي وسائط مثل الدالة ( ) Now في حين أي دالة Function لا بد وأن يكون لها ناتج. ○

ن الإجراءات Sub لا يجوز استخدامها في أي جملة تخصيص.

ن إجراء الحدث Event Procedure يعتبر إجراء من النوع Sub.

ن وسيط الإجراء Procedure إما أن يكون قيمة مجردة أو متغير Variable أو ثابت Constant أو دالة Function

**دالة ( ) MsgBox**

**MessageBox تصنيف Class**

**( ) Show دالة معلنة داخل التصنيف MessageBox**

## الفصل الخامس التعدي الإلكتروني Cyber bullying

### أولاً : مقدمة :

إن أخلاقيات التعامل مع الإنترنت بل وكافة وسائل المعلومات والاتصالات أصبحت من موضوعات الساعة التي تهتم الأفراد والمجتمعات والدول لذا نسعى في هذا الفصل إلى رفع الوعي لدى أبنائنا وبناتنا بأهمية أخلاقيات التعامل مع الإنترنت واكسابهم المعلومات والمهارات اللازمة لرفع درجة سلامتهم الشخصية فيما يتعلق بالتعدي عبر الإنترنت . ومن خلال الإنترنت نستطيع أن نتعلم و نتثقف و نتسلى و نتواصل و نتحاور ولكن يمكن أن نحصل منها على معلومات خطأ ويمكن أن نتعرف على أشخاص سيئين.

**تعريف التعدي الإلكتروني :** عبارة عن سلوك عدواني متعمد يستخدم الوسائط الإلكترونية بغرض :

- ١ . التحرش .
- ٢ . المضايقة .
- ٣ . الإحراج .
- ٤ . التخويف .
- ٥ . التهديد .

**ثانياً : الوسائط الإلكترونية للتعدي :** الوسائط الإلكترونية التي يستخدمها المعتدي الإلكتروني كثيرة منها ما يلي :

- ١ . البريد الإلكتروني E-Mail : من خلاله يمكنك إرسال واستقبال الرد وإعادة توجيه الرسائل الكترونياً مع إمكانية إرفاق ملفات بتلك الرسائل .
- ٢ . المنتديات الإلكترونية Forums : فيها يتم طرح موضوعات للمناقشة فيتم إرسال المشاركة وجد الردود من الآخرين .
- ٣ . الرسائل الفورية Instant Message : من خلالها يحدث اتصال فوري بين طرفين أو أكثر عن طرق الكتابة أو المحادثة الصوتية أو المرئية .
- ٤ . التدوين الإلكتروني Bloging : عبارة عن سجل الإلكتروني خاص يدون به صاحب المدونة ما يشاء .
- ٥ . المواقع الاجتماعية مثل FaceBook : تسمح بنشر معلومات خاصة للشخص صاحب الحساب .

### ثالثاً : أشكال التعدي الإلكتروني :

- ١ . التخفي الإلكتروني (Anonymity) : عبارة عن " استخدام اسماء مستعارة تخفي شخصية المعتدي الإلكتروني بغرض الإفلات من العقاب " .
- ٢ . المضايقات الإلكترونية (Harassment) : عبارة عن " رسائل عدائية ومزعجة موجهة ضد شخص أو أكثر " .
- ٣ . الملاحقة الإلكترونية (Cyber stalking) : هي " شكل من أشكال المضايقات الإلكترونية لكن بشكل متكرر " .
- ٤ . السب أو القذف الإلكتروني (Flaming) : عبارة عن " نشر تعليقات أو كلمات عدائية و مبتذلة ضد شخص أو أكثر " .
- ٥ . التشهير الإلكتروني (Outing) : عبارة عن " نشر معلومات عن شخص محدد أو أكثر بشكل مسيء " .
- ٦ . الاستثناء الإلكتروني (Exclusion) : عبارة عن " تجاهل شخص أو أكثر من خلال وسائط إلكترونية " .
- ٧ . التهديد الإلكتروني (Threats Cyber) : عبارة عن " إرسال رسائل إلكترونية تحمل تهديد أو وعيد لشخص أو أكثر " .



## كيف تحمي نفسك من التعدي الإلكتروني؟

٨. لا تشارك أحد بكلمة السر، مهما كان قريب منك (كصديق) ، وأن تتعامل مع كلمة السر كمفتاح شفتك لا تعطيه لأي صديق من أصدقائك.
٩. إعداد كلمة مرور لا تستنتج ، وذلك باستخدام مزيج من الحروف والأرقام "غير متعلقة بإسمه أو سنة ميلاده أو تفضيلاته أو أبناءه أو الأشخاص المقربين جداً منه".
١٠. عدم نشر أى بيانات خاصة أو عائلية على الإنترنت، بغض النظر عن طريق البريد الإلكتروني، أو النصوص المكتوبة، أو اليوميات الإلكترونية، أو على المواقع الإلكترونية.
١١. عدم حذف رسائل التعدي الإلكتروني ، حيث يمكن أن تحتاجها كأدلة ضده.
١٢. عدم مقابلة أحدا تعرفت عليه من خلال الإنترنت.
١٣. عدم إرسال رسائل وأنت في حالة غضب.
١٤. إطلاع ولي الأمر بما يضايقك عند استخدام الإنترنت أو من تثق بهم من كبار العائلة أو من المعلمين إذا تلقيت بريد أو رسالة الكترونية تجعلك تشعر بعدم الراحة، وأخبر وشارك أحد الكبار بتجاربك على شبكة الإنترنت.
١٥. إنزال البرامج من الإنترنت يكون تحت إشراف معلمك أو ولي أمرك.
١٦. قم بالإبلاغ عن التعدي الإلكتروني للسلطات المختصة.
١٧. لا ترد على المتعدي إلكترونياً، ولا تصدق كل ما يكتبه على الإنترنت والانتظار حتى تهدأ.
١٨. احترم دائماً حق الآخرين على شبكة الإنترنت، وتعلم الخلق الحسن الذي يعتبر فن أدب الحوار على شبكة الإنترنت.

## مواقفهم توضح الإرشادات السابقة

- ١ - قال عمرو لياسمين أريد أن أرسل رسالة لصديق وليس عندي حساب بريد الكتروني هل من الممكن أن تعطيتني اسم المستخدم وكلمة المرور الخاصين بك حتى أتمكن من إرسال هذه الرسالة. في رأيك ماذا تفعل ياسمين ؟  
الإجابة : لا تشارك أحد بكلمة السر.
- ٢ - قام عمرو بإنشاء حساب بريد الكتروني خاص به فحاولت ياسمين استنتاج كلمة المرور بأن كتبت اسمه ثم سنة ميلاده ففتح الحساب. في رأيك ماذا كان يجب أن يفعل عمرو ؟  
الإجابة : مراعاة إعداد كلمة مرور لا تستنتج.
- ٣ - قام رامي بالإشتراك في خدمة الـ Facebook وقام بنشر معلومات تخص أسرته مستعرضاً صور الأسرة وأنشطتهم اليومية فقام لص بدراسة تلك المعلومات وعرف متى يكون المنزل غير متواجد به أحد ثم .....؟!  
في رأيك ماذا كان يجب أن يفعل رامي ؟  
الإجابة : عدم نشر أى بيانات خاصة أو عائلية على الإنترنت.
- ٤ - اشتكى أحد الطلاب لمعلمه أن زميله سبه في رسالة بريد الكتروني فقال المعلم أرني تلك الرسالة ، فقال الطالب لقد حذفها فقال المعلم ضيعت الدليل. في رأيك ماذا كان يجب أن يفعل الطالب ؟  
الإجابة : عدم حذف رسائل التعدي الإلكتروني ، حيث يمكن أن تحتاجها كأدلة ضده.
- ٥ - قال عمرو لوالده لقد تعرفت على شخص في إحدى غرف المحادثة ويود أن يقابلني في رأيك ماذا كان رد والده ؟  
الإجابة : عدم مقابلة أحدا تعرفت عليه من خلال الإنترنت.
- ٦ - حدثت مشاحنة كبيرة بين طالبين وبعد أن انتهت كان أحدهما غاضباً جداً فبعث رسالة تهديد ووعيد للآخر. في رأيك ماذا كان يجب أن يفعل الطالب ؟  
الإجابة : يجب أن ينتظر حتى يهدأ ، و عدم إرسال رسائل وأنت في حالة غضب.